International Journal of Control, Automation and Systems 18(2) (2020) 322-329
http://dx.doi.org/10.1007/s12555-019-0256-5

ISSN:1598-6446 eISSN:2005-4092
http://www.springer.com/12555

# Error Improvement in Visual Odometry Using Super-resolution

**Wonyeong Jeong*, Jiyoun Moon, and Beomhee Lee**

**Abstract:** Visual odometry (VO), a method that estimates odometry using visual sensors, is hard to operate successfully with the low-resolution and noisy image sequences. To address this problem, a super-resolution technique is applied to input data before performing VO. Since most conventional super-resolution literature mainly deals with the resolution increment, we present a novel deep neural super-resolution network, which can remove noises as well. The execution time is also taken into account by adjusting the number of CNN layers for a real-time VO. By applying the proposed super-resolution approach, the resolution increases and noises disappear with a suitable speed, hence VO can be performed successfully. Experimental results show that the proposed method improves the VO performance compared with the conventional VO which uses low-resolution and noisy image sequences.

**Keywords:** Robust Visual Odometry, super-resolution, visual odometry, visual SLAM.

## 1. INTRODUCTION

In robotics, robust odometry estimation is essential for the robot autonomy. To find the odometry of a robot, various algorithms were introduced by combining one or several sensor information. Among various sensors, visual ones are being actively used because they can provide rich information about the environment at the low-cost. The technique for estimating odometry using only RGB cameras is called visual odometry (VO); it is called monocular VO if only one camera is used. VO has been studied actively in robotics and computer vision fields, and it has begun to be utilized to various application, such as unmanned aerial vehicle control, 3D modeling, augmented reality, and autonomous driving cars.

Since VO utilizes only cameras, the performance of the camera and the quality of images greatly affect the result. Although plenty of image sequence datasets for VO research exist online, most of them are acquired by expensive high-resolution (HR) and low-noise cameras. In order to adopt VO in various applications, it is necessary to maintain their performance even if low-resolution (LR) and noisy cameras, which are often equipped in mobile platforms, are used. However, when using an LR and noisy image sequence, the performance of VO is remarkably reduced as displayed in Fig. 1. In VO result of using LR and noisy images, tracking procedure is lost while a camera moves, which leads to a catastrophic failure.

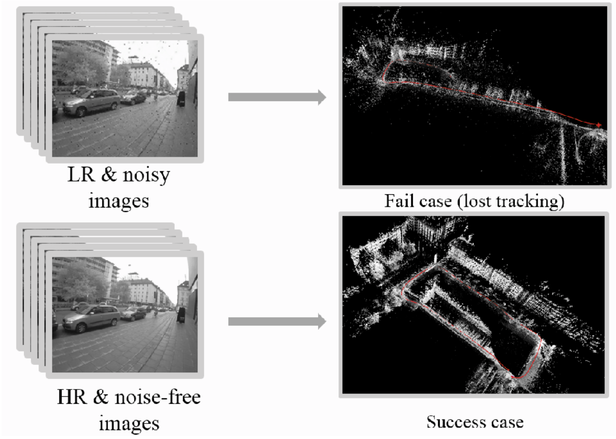


Fig. 1. Comparison of results of VO using LR and noisy with HR and noise-free image sequences. Red lines in right images denote paths of cameras.

In this paper, we exploit a super-resolution (SR) technique to convert an LR and noisy image into an HR and low-noise image for achieving a successful VO. Among various SR approaches, the deep learning-based one, which has recently shown better performance, is adopted. Conventional SR structures are challenging to apply this paper because of two main problems. The first problem is the excessive execution time due to the too deep network. The second problem arises from the poor noise removal performance that is owing to their network structure. Therefore, we propose a new deep neural SR architecture that can achieve the low-error and real-time VO.

Manuscript received April 3, 2019; revised June 24, 2019; accepted September 6, 2019. Recommended by Associate Editor Hyun Myung under the direction of Editor Won-jong Kim. This work was supported in part by the Brain Korea 21 Plus Project and in part by a Bio–Mimetic Robot Research Center funded by Defense Acquisition Program Administration, and by Agency for Defense Development (UD190018ID).

Wonyeong Jeong, Jiyoun Moon, and Beomhee Lee are with Automation and Systems Research Institute, Department of Electrical Engineering, Seoul National University, Seoul, Korea (e-mails: {jwyeong, jiyounmoon, bhlee}@snu.ac.kr).
* Corresponding author.

Experimental results show that the performance of VO using SR image sequences is better than that of the conventional methods using LR and noisy image sequences.

The rest of this paper is organized as follows: Section 2 looks at related work of VO and SR. Section 3 describes the process of finding a resolution suitable for VO through experiments and the proposed network structure of SR. Section 4 summarizes the experimental results and their analysis. Finally, Section 5 concludes this paper and discusses the future work.

## 2. RELATED WORK

### 2.1. Visual odometry

Estimating odometry using a visual sensor had been studied previously, but the word *visual odometry* was only coined by Nister *et al.* [1] in 2004. VO can be divided into the feature-based method and the direct method. Feature-based methods utilize the feature extraction and the feature matching, which were the main stream in the early VO research. Initially, feature locations and camera poses of all frames were estimated by filters [2–4]. These approaches caused too much computation while little new information was obtained, since consecutive frames were frequently captured in the immediate vicinity. To alleviate this problem, PTAM [5] estimated poses of the chosen frames, namely keyframes. Moreover, it dealt with tracking and mapping in parallel threads, and enabled real-time VO successfully in small environments. Today, the most representative feature-based literature is probably ORB-SLAM [6, 7]. The feature used in [7] was ORB, which is based on FAST, and it can be extracted and matched faster than those of using SIFT or SURF. [7] exploited the same feature in all SLAM tasks of its framework - tracking, mapping, relocalization and loop-closing, which resulted in more efficient, simple, and reliable system than the conventional methods.

The feature-based VO is robust to various problems caused in the image acquisition process, such as an automatic exposure change, a non-linear response function, lens attenuation and even a rolling shutter effect. However, in low-texture areas, such as simple corridors or walls, feature extraction is difficult to achieve and this leads to the failure of estimating odometry.

In direct approaches, pixel intensities are used directly rather than features. Direct methods warp pixels from one image to another, and then obtain a transformation between images that minimizes the sum of intensity differences. DTAM [9] and REMODE [10] optimized the whole pixels to perform VO densely, thus they were hard to achieve the real-time execution except powerful GPU devices. To reduce this computational burden, Schops *et al.* [11] proposed a semi-dense manner which used pixels with high intensity gradient. Based on [11], Engel *et al.* [12] proposed LSD-SLAM that performed visual SLAM in real-time using single CPU in a large scale environment. Furthermore, Engel *et al.* [13] proposed a direct sparse odometry (DSO), the state-of-the-art direct VO literature, using more sparse pixels than LSD-SLAM. DSO improved the performance of VO by estimating the exact pixel intensity value considering the exposure time, the response functions and the lens attenuation of the image.

The direct VO, since intensities of the pixel are directly used, is performed based on more information than feature-based methods, hence the algorithm can be performed well in the low-texture region. However, because it uses low-level information, it is vulnerable to distortions which easily arise from the image acquisition process.

### 2.2. Super-resolution

SR is the one of image restoration techniques that generates an HR image from an LR image. Initially, SR was done by simple interpolation using sampling theories [14, 15], however these approaches were difficult to predict the detailed parts of an image. As an improvement, methods of learning a function that matches a pair of LR image and HR image were presented. These methods include neighbor embedding [16, 17] and sparse coding [18, 19]. Similarly, learning the transformation of patches using internal similarity [20, 21] were proposed.

Recently, SR research has made great progress in performance by employment of deep learning techniques. Dong *et al.* [22, 23] proposed the first work to introduce the idea of applying a convolutional neural network (CNN) [24] to SR. Their method, named SRCNN, conducted an SR in an end-to-end manner utilizing a CNN network which consisted of three convolutional layers. However, the shallow network converged slowly and had not been able to learn many nonlinearities. VDSR [25, 26] claimed that the deeper network makes the better image quality and used twenty convolutional layers to improve the SR performance. Also, VDSR added the input image to the output of the last layer to train the residual only, which made the convergence time shortened. Around the same time, He *et al.* [27] proposed ResNet that performed well in the classification and the detection, which are other computer vision fields, by learning the residual in the middle of the network. Using ResNet structure and a generative adversarial network, Ledig *et al.* [28] proposed SR-ResNet. However, ResNet was not an optimal structure for SR since it was designed for different purposes. Therefore, EDSR [29] removed unnecessary modules in ResNet structure, which led to advancement in performance.

Although SR algorithms mainly focus on increasing the resolution, removing noise is also considered in the proposed SR network. Furthermore, the time elapsed for passing deep network is taken into account since we intend to combine SR with VO.

## 3. SR-VO

In this paper, we have improved the direct VO, whose performance is more sensitive to the image quality than that of the feature-based VO. All VO used in experiments for this paper is DSO, the state-of-the-art algorithm among direct algorithms.

### 3.1. DSO performance analysis according to changing resolution

The performance of VO highly depends on the image resolution. Apparently, the higher the resolution, the better the VO performance, but the number of addressable frames per second (fps) also decreases. Therefore, it is necessary to find the optimal resolution with a smaller error while guaranteeing an appropriate fps. To find this resolution, each sequence of the TUM dataset [30] was tested five times with various resolutions for analyzing time and error. The error metric utilized is root mean square error (RMSE). Since the scale and the direction are changed every time VO is executed, the estimated and the ground truth poses must be adjusted before calculating RMSE. Let poses be $\mathbf{p}_i = \{x_i, y_i, z_i\}$ for $i = 1 \cdots n$ at timestep $i$. Estimated and the ground truth poses are then represented by ${}^{est}\mathbf{p}_i$ and ${}^{gt}\mathbf{p}_i$, respectively. The direction (rotation), the origin, and the relative scale are factors that have to be aligned before computing RMSE. First, the rotation matrix is calculated by using singular value decomposition as follows:

$$R = UV', \tag{1}$$

where $U$ and $V$ are orthogonal matrices composed of singular vectors of the cross-covariance of $\{{}^{est}\mathbf{p}_i\}$ and $\{{}^{gt}\mathbf{p}_i\}$. Next, to identify the origins, new poses are set as follows:

$$\begin{aligned} {}^{est}\mathbf{p}'_i &= ({}^{est}\mathbf{p}_i - E[{}^{est}X])R, \\ {}^{gt}\mathbf{p}'_i &=^{gt} \mathbf{p}_i - E[{}^{gt}X], \end{aligned} \tag{2}$$

where $E[X]$ is the expectation of $X$. The relative scale $s = s_{gt}/s_{est}$, the last alignment, is recovered by following equation:

$$s = \frac{\sum_{i=1}^{n} \|{}^{gt}\mathbf{p}_i\|}{\sum_{i=1}^{n} \|{}^{est}\mathbf{p}_i\|}. \tag{3}$$

Finally, RMSE is calculated as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \|{}^{est}\mathbf{p}'_i \cdot s -^{gt} \mathbf{p}'_i\|^2}. \tag{4}$$

Obtained by above manner, RMSEs of various resolutions and sequences are displayed in Fig. 2.

From the figure, the lower the resolution, the higher the RMSE value, as expected. Note that RMSE of back sequences (after 17) are smaller than those of front sequences. This is because the configuration of the TUM dataset. Previous sequences are typically captured in indoor environments with small scale rooms and corridors, and sequences after 17 are either large scale indoor (with high ceilings and lobby) or outdoor environments. Also, the first part of the sequence is a complex path, while the second part is a simple path that makes one large loop. This difference can be attributed to the difference in the RMSE. The execution time and RMSE changes with the resolution are shown in Fig. 3. The ratio of the width to the height of the image is 4:3 and *x-axis* in Fig. 3 denotes width of a image. From the left graph in Fig. 3, RMSE value decreases drastically at low resolution but gradually converges and eventually makes no big difference after 704 × 528. In the case of time, the average time of conducting VO increases as the resolution increases. Therefore, using 704 × 528 resolution, we can confirm that VO can be performed at 23.67fps with low error. Hence, in the remainder of this paper, we experimented with SR learned to 704 × 528.

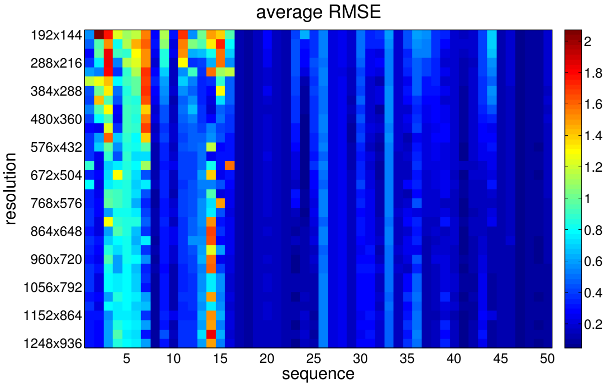


Fig. 2. RMSE of various resolutions of all sequences. One grid means the average of five repeated experiments.

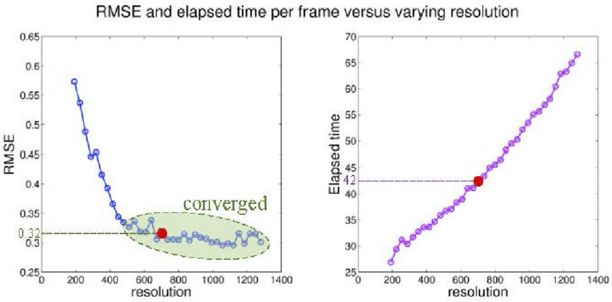


Fig. 3. RMSE (left) and elapsed time per frame (right) variation with resolution changes. Red dot denotes chosen suitable resolution for VO.

### 3.2. Super-resolution network

Among various SR approaches, some algorithms can increase the input resolution to arbitrary values, but others

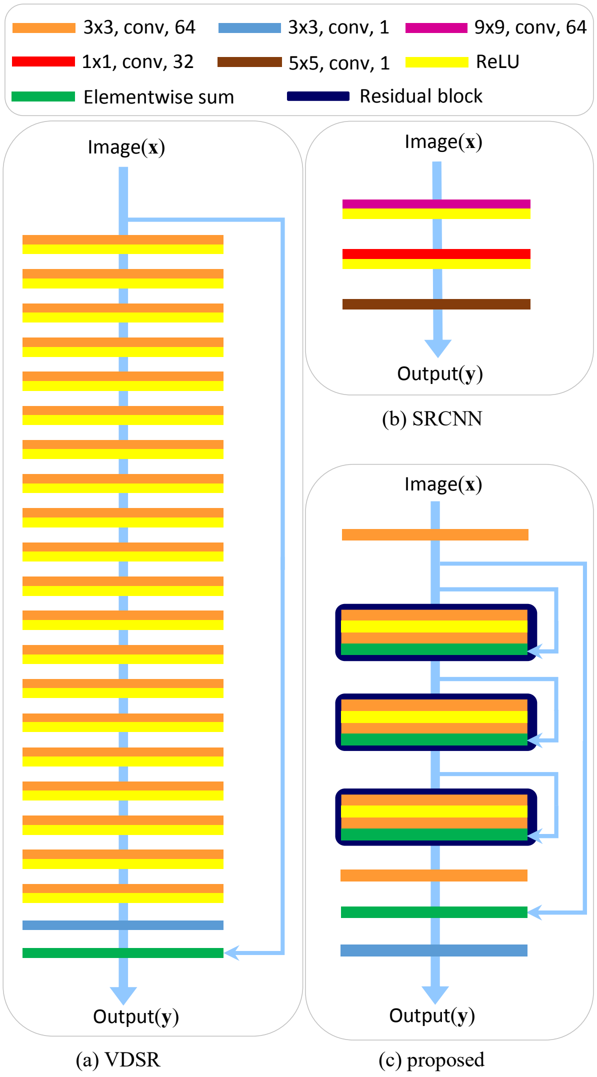


Fig. 4. Comparison of super-resolution networks of VDSR, SRCNN, and the proposed

can acquire only integer multiples of the input resolution. The former methods upsample to the desired resolution by the bicubic interpolation and then pass CNN networks to get an SR image [22, 25]. In the latter methods, upsampling processes are in the middle of the network hence they cannot obtain the arbitrary resolution [28, 29]. In this paper, we need to perform SR with 704 × 528 resolution from arbitrary LR image. Therefore, methods which cannot acquire arbitrary output resolutions like EDSR and SRResNet are not appropriate for our algorithm, thus SRCNN and VDSR are only applied. Network structures of SRCNN, VDSR and the proposed are depicted in Fig. 4.

As shown in figure, SRCNN and VDSR have three and twenty convolutional layers, respectively. We first analyze the noise removing property of two conventional networks. SRCNN has too shallow CNN layers to get rid of noises. In VDSR, the input image is added to the image that passed the last convolutional layer. This enables the network to learn only high-frequency parts hence improves the performance and accelerates the convergence. However, since the input image is added at the end, noises contained in the input are not totally eliminated.

Computation times are concerned with the number of operations in SR networks. Assuming equal input sizes, operation numbers of convolutional layers are proportional to $f_i \times k \times f_o$, where $f_i$, $f_o$, $k$ are input and output feature numbers and the kernel size, respectively. Calculating the number of operations of SR networks in this manner, SRCNN and VDSR are operated with about $8.1k$ and $660k$ operations, respectively; these numbers are reflected in the runtime of algorithms. Comparing the execution times of the two algorithms, the SRCNN operates at 144 fps with an average of 6.92ms per frame and the VDSR operates at 19.5 fps with an average of 51.4ms per frame. The execution time of VDSR is slower than that of DSO, which means that real-time VO with SR is unavailable. On the other hand, SRCNN is faster than enough so, more convolutional layers could be added.

As a result, we design SR using nine convolutional layers, which operates with about $260k$ operations. The proposed network adds a convolutional layer at the beginning and the end to prevent direct propagation of noise from the input image to the output. We constructed the network using residual blocks, and constant scaling is applied to each residual block. The equation presents the residual blocks as follows:

$$Res(\mathbf{x}) = c(W_1\sigma(W_1\mathbf{x}) + \mathbf{x}), \tag{5}$$

where $\mathbf{x}$ and $Res(\mathbf{x})$ are the input and the output of the residual block, $c$ is a scaling constant, $\sigma$ and $W_1$ denote the ReLU function and a convolutional layer with $3 \times 3$ kernel sizes and 64 filters, respectively. The total equations of the proposed network are as follows:

$$\mathbf{y} = W_2(W_1 Res^3(W_1\mathbf{x}) + W_1\mathbf{x}), \tag{6}$$

where $W_2$ denotes a convolutional layer with $3 \times 3$ kernel size and 1 filter.

## 4. EXPERIMENTS

### 4.1. Super-resolution procedure

#### 4.1.1 Training SR

In this paper, we used the monocular visual odometry dataset [30] provided by Technical University of Munich. This dataset consists of 50 sequences, including indoor and outdoor environment and the number of total images is 190,576. To perform SR, ten sequences (5, 10, ..., 50) in multiples of 5 were set as the test data, and the remaining 40 sequences were used as the training data. The number of the training images is 154,256 and that of the test is 36,320. The original resolution of this dataset is

1280 × 1024, so 704 × 528, which is the ground truth resolution of SR, is obtained by using bicubic downsampling. In training data, the resolution of the original image was downsampled to 320 × 240 and 192 × 144. Therefore, the scale of SR is 2.2 and 3.6563, respectively. Furthermore, to make noisy images, salt and pepper noises are added on downsampled images with 0.4% of whole pixels.

Training details are as follows: The image patch used in the training was a 44 × 44 grayscale image. The Adam optimizer the $\mathcal{L}_2$ function were used as the optimizer and the Loss function, respectively. The batch sizes were 32, 4, and 8 for SRCNN, VDSR, and the proposed method, respectively, depending on the memory capacity of the graphics card. The initial learning rate was set to $10^{-4}$, and it was divided by 10 after every 10 epochs. All methods were trained until convergence. The epochs required for learning were SRCNN of 60, VDSR and the proposed network of 40. We configured methods as python language and utilized NVIDIA GTX 1080 Ti GPU.

### 4.1.2 Testing SR

We tested the proposed networks on the part of the TUM monocular dataset. We compared our method with bicubic, SRCNN, and VDSR. For SRCNN and VDSR, we utilized our own learning outcomes. Table 1 shows a quantitative result that presents average of peak signal-to-noise ratio (PSNR), structural similarity (SSIM) and computation time of SR methods.

Table 1. Average PSNR, SSIM and time results of Super-resolutions.

| | | PSNR | SSIM | time (ms) |
|---|---|---|---|---|
| 320 × 240 images | bicubic | 30.00 | 0.856 | 0.646 |
| | SRCNN | 32.62 | 0.892 | 5.633 |
| | VDSR | 32.94 | 0.899 | 51.14 |
| | proposed | **34.41** | **0.913** | 27.18 |
| 192 × 144 images | bicubic | 27.95 | 0.808 | 0.463 |
| | SRCNN | 29.46 | 0.832 | 8.208 |
| | VDSR | 29.61. | 0.844 | 51.58 |
| | proposed | **30.93** | **0.856** | 26.26 |

The proposed method shows the best performance in both PSNR and SSIM, followed by VDSR, SRCNN, and bicubic in order. In terms of time, the average elapsed time of VDSR per image is the longest, 51.36ms, which is slower than processing speed of DSO. Bicubic and SRCNN was maintained at over 100fps, and the proposed method showed a speed at about 37fps. The bicubic method is faster than other methods since it simply interpolates neighboring pixels to increase the resolution unlike using a deep neural network. Results of CNN-based SR methods are shown in accordance with the operation numbers of networks as analyzed in section 3.2. For qualitative comparison, one SR result is shown in Fig. 5. We zoomed in and compared one car included in images.

As seen left images in the table of Fig. 5, results of learning-based methods show more keen boundary than that of the bicubic interpolation. In noise removing, the noises of the bicubic result are not eliminated and rather the size is expanded, since it is an interpolation method. Also, noises are not completely removed in the SRCNN and VDSR results, whereas the proposed method eliminates almost all noises.

In right images in the table of Fig. 5, which are results of SRs using 192 × 144 images. The result of the bicubic shows much larger noises than that of the 320 × 240. Results of learning-based models are seemed sharp car contours, but noises remain in outcomes of SRCNN and VDSR.

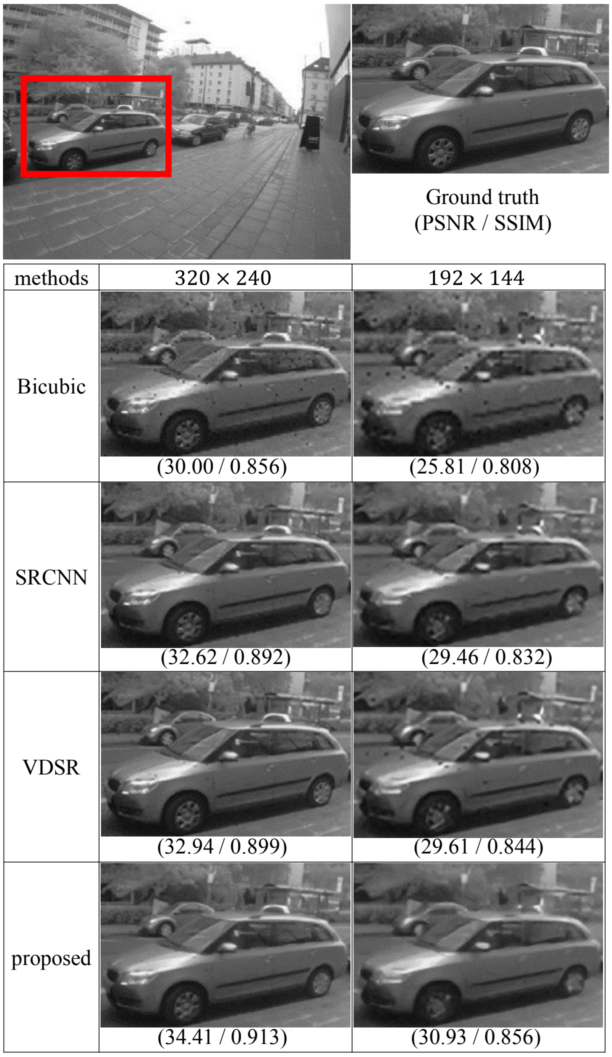


Fig. 5. Super-resolution outcomes of bicubic, SRCNN, VDSR and the proposed; A left top image is the ground truth image and the rest images are zoomed parts of the red box.

Table 2. Results of the RMSE and the fps processed by DSO and SR.

| | | seq. 5 | seq. 10 | seq. 15 | seq. 20 | seq. 25 | seq. 30 | seq. 35 | seq. 40 | seq. 45 | seq. 50 | avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LR image | RMSE | 1.214 | 0.427 | 1.934 | 0.343 | 0.459 | 0.491 | 0.392 | **0.191** | 0.134 | 0.199 | 0.578 |
| (320 × 240) | fps(Hz) | 40.16 | 40.92 | 27.27 | 30.08 | 30.70 | 27.60 | 33.70 | 33.77 | 30.34 | 28.51 | 32.31 |
| bicubic + | RMSE | 1.209 | 0.505 | 1.894 | 0.347 | 0.485 | 0.509 | 0.494 | 0.195 | 0.133 | 0.196 | 0.597 |
| 320 × 240 | fps(Hz) | 34.75 | 33.10 | 23.89 | 26.46 | 26.02 | 24.27 | 30.54 | 26.77 | 25.46 | 23.94 | 28.61 |
| SRCNN + | RMSE | 1.200 | 0.487 | 1.746 | 0.341 | 0.441 | 0.491 | 0.413 | 0.195 | 0.128 | 0.198 | 0.564 |
| 320 × 240 | fps(Hz) | 37.82 | 36.12 | 26.85 | 29.87 | 29.52 | 27.78 | 33.54 | 29.00 | 28.92 | 26.43 | 30.59 |
| VDSR + | RMSE | 1.184 | 0.224 | **1.503** | 0.343 | 0.388 | 0.502 | 0.414 | 0.192 | 0.138 | 0.196 | 0.509 |
| 320 × 240 | fps(Hz) | 13.81 | 13.58 | 11.88 | 12.34 | 12.23 | 11.90 | 13.15 | 12.35 | 12.18 | 11.90 | 13.06 |
| proposed + | RMSE | **0.803** | **0.152** | 1.658 | **0.118** | **0.109** | **0.250** | **0.129** | 0.191 | **0.117** | **0.136** | **0.366** |
| 320 × 240 | fps(Hz) | 21.26 | 20.63 | 16.63 | 17.84 | 17.64 | 16.82 | 19.60 | 17.98 | 17.38 | 16.66 | 18.24 |
| LR image | RMSE | 1.105 | 1.047 | 1.874 | 0.343 | 0.604 | 0.508 | 0.456 | 0.193 | 0.156 | 0.185 | 0.647 |
| (192 × 144) | fps(Hz) | 44.81 | 44.37 | 29.21 | 33.08 | 36.45 | 32.33 | 43.14 | 36.97 | 38.94 | 39.10 | 37.84 |
| bicubic + | RMSE | 1.125 | 3.885 | 1.953 | 0.347 | 0.598 | 0.499 | 0.506 | 0.197 | 0.148 | **0.178** | 0.943 |
| 192 × 144 | fps(Hz) | 35.54 | 33.90 | 25.42 | 27.44 | 27.42 | 25.20 | 30.15 | 27.10 | 27.37 | 26.60 | 27.52 |
| SRCNN + | RMSE | **1.070** | 4.005 | 1.927 | 0.343 | 0.607 | 0.490 | 0.514 | 0.193 | 0.149 | 0.181 | 0.948 |
| 192 × 144 | fps(Hz) | 38.62 | 36.85 | 28.13 | 30.15 | 30.51 | 28.46 | 33.96 | 30.12 | 30.99 | 29.21 | 31.70 |
| VDSR + | RMSE | 1.209 | 0.633 | 1.976 | 0.345 | **0.472** | 0.515 | 0.493 | 0.194 | 0.146 | 0.200 | 0.618 |
| 192 × 144 | fps(Hz) | 14.32 | 14.03 | 12.45 | 12.86 | 12.98 | 12.59 | 13.29 | 12.58 | 12.80 | 12.61 | 12.53 |
| proposed + | RMSE | 1.172 | **0.604** | **1.586** | **0.333** | 0.550 | **0.336** | **0.379** | **0.187** | **0.137** | 0.191 | **0.548** |
| 192 × 144 | fps(Hz) | 21.55 | 20.94 | 17.36 | 18.28 | 18.27 | 17.26 | 19.44 | 18.13 | 18.25 | 17.90 | 18.74 |

### 4.2. VO with SR images

Experiments were carried out for two resolutions of 320 × 240 and 192 × 144 and for four SR methods - bicubic, SRCNN, VDSR, and the proposed - compared with VO using LR images. RMSE and the frequency variations of each method are shown in Table 2. Note that the bicubic method produces worse result than the LR image. This is a problem with the interpolation method. The bicubic method interpolates pixel intensities when conducting upsampling, which results in the effect of smoothing the image. In DSO, the optimization is performed using pixels with high intensity gradient. Hence, when the image is smoothed, the gradient becomes low and the number of available pixels is reduced. Therefore, bicubic interpolation can be seen as inappropriate when performing VO.

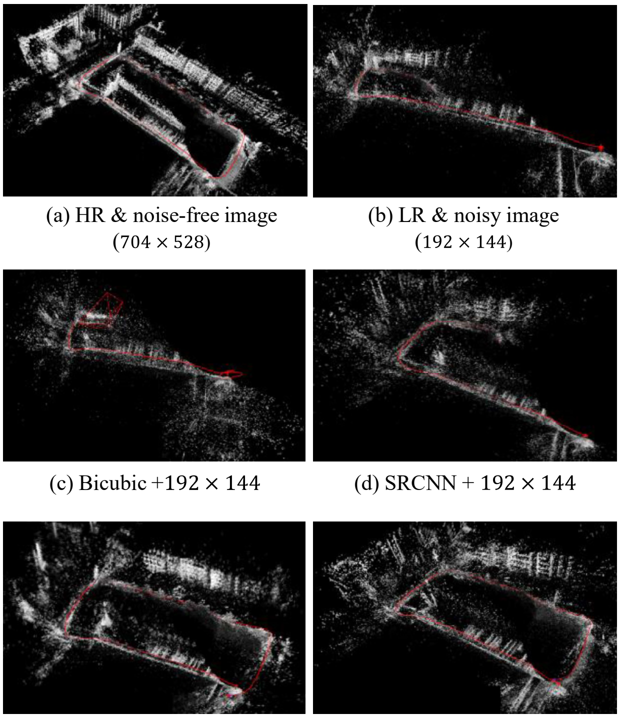
On the other hand, since learning-based SRs restore the detailed part of the image, they show better VO performance. Overall, the results of the proposed method showed lowest RMSE, but other methods were better in a few sequences. This is because uncertainties happened in the process of choosing pixels utilized in optimization. If the number of pixels above the gradient threshold exceeds the designated maximum number, arbitrary pixels are chosen thus randomness occur.

In the frequency aspect, the bicubic and SRCNN are faster than even using HR images directly. This is because pixels used in DSO is less when using the bicubic and SRCNN SR images, resulting in optimization process shortened. The proposed method showed about 18fps in both resolutions which is five fps lower than frequency using HR images. As a result, the result of the proposed method is the best performance in RMSE and is suitable for real-time VO. The qualitative VO result is shown in Fig. 6. Out of the ten test sequences, the ninth sequence (sequence 45) is presented.

The sequence 45 was collected from an outside environment, whose path consists of turning around a building, then returning to the starting point. The result of using HR and noise-free image sequence showed that both the path and the reconstruction are clean. On the other hand, using LR and noisy image sequence caused tracking lost as shown in Fig. 6(b). The result utilizing the bicubic method suffered from scale problem and eventually lost tracking. In SRCNN result, the scale was wrongly estimated when going out to the lobby, so that odometry was totally misjudged. As shown in Fig. 6(e), the result of using the VDSR showed similar to Fig. 6(a), but a little skew of the path existed and failed to return to the same location. Finally, the result of the proposed method showed quite similar output with that of using HR and noise-free images, though reconstruction points spread widely.

## 5. CONCLUSION AND FUTURE WORK

In this paper, we propose a method to improve the low-performance of VO when using LR and noisy image sequences. We designed an SR network that deals with noises and execution time as well as resolution increment differently from other SR techniques. The proposed SR makes the image quality increase, which leads to a suc-



(a) HR & noise-free image (704 × 528)

(b) LR & noisy image (192 × 144)

(c) Bicubic +192 × 144

(d) SRCNN + 192 × 144

(e) VDSR + 192 × 144

(f) Proposed method + 192 × 144

Fig. 6. DSO results of using HR images, LR images, and SR images which are upsampled by bicubic, SRCNN, VDSR, and the proposed method.

cessful VO result. Experimental results show that the performance of the proposed method is better than that of conventional VO. This work can be utilized to real applications, such as augmented reality applications and the autonomous driving since VO performs well even when a low-cost camera is used.

## REFERENCES

[1] D. Nistèr, O. Naroditsky, and J. Bergen, "Visual odometry," *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. II, 2004.

[2] E. Eade and T. Drummond, "Scalable monocular SLAM," *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 469476, 2006.

[3] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: real-time single camera slam," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, no. 6, pp. 10521067, 2007.

[4] J. Civera, A. J. Davison, and J. M. Montiel, "Inverse depth parametrization for monocular SLAM," *IEEE Trans. on Robotics*, vol. 24, no. 5, pp. 932945, 2008.

[5] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," *Proc. of IEEE and ACM International Symposium on Mixed and Augmented Reality (ASMAR)*, pp. 225234, 2007.

[6] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: an open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. on Robotics*, vol. 33, no. 5, pp. 12551262, 2017.

[7] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: a versatile and accurate monocular slam system," *IEEE Trans. on Robotics*, vol. 31, no. 5, pp. 11471163, 2015.

[8] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: an efficient alternative to SIFT or SURF," *Proc. of IEEE International Conf. Computer Vision (ICCV)*, pp. 25642571, 2011.

[9] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: dense tracking and mapping in real-time," *Proc. of IEEE International Conf. Computer Vision (ICCV)*, pp. 23202327, 2011.

[10] M. Pizzoli, C. Forster, and D. Scaramuzza, "REMODE: probabilistic, monocular dense reconstruction in real time," *Proc. of IEEE International Conf. Robotics and Automation (ICRA)*, pp. 26092616, 2014.

[11] J. Engel, J. Sturm, and D. Cremers, "Semi-dense visual odometry for a monocular camera," *Proc. of IEEE International Conf. on Computer Vision (ICCV)*, pp. 14491456, 2013.

[12] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: large-scale direct monocular SLAM," *Proc. of the European Conference on Computer Vision*, pp. 834849, 2014.

[13] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 40, no. 3, pp. 611625, 2018.

[14] X. Li and M. T. Orchard, "New edge-directed interpolation," *IEEE Trans. on Image Processing*, vol. 10, no. 10, pp. 15211527, 2001.

[15] L. Zhang and X. Wu, "An edge-guided image interpolation algorithm via directional filtering and data fusion," *IEEE Trans. on Image Processing*, vol. 15, no. 8, pp. 22262238, 2006.

[16] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi-Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," *Proceedings of British Machine Vision Conference*, pp. 135.1-135.10, 2012.

[17] X. Gao, K. Zhang, D. Tao, and X. Li, "Image super-resolution with sparse neighbor embedding," *IEEE Trans. on Image Processing*, vol. 21, no. 7, pp. 31943205, 2012.

[18] J. Yang, Z. Wang, Z. Lin, S. Cohen, and T. Huang, "Coupled dictionary training for image super-resolution," *IEEE Trans. on Image Processing*, vol. 21, no. 8, pp. 34673478, 2012.

[19] R. Timofte, V. De Smet, and L. Van Gool, "A+: adjusted anchored neighborhood regression for fast super-resolution," *Proc. of Asian Conf. on Computer Vision (ACCV)*, pp. 111126, 2014.

[20] G. Freedman and R. Fattal, "Image and video upscaling from local self-examples," *ACM Trans. on Graphics (TOG)*, vol. 30, no. 2, p. 12, 2011.

[21] Z. Wang, Y. Yang, Z. Wang, S. Chang, J. Yang, and T. S. Huang, "Learning super-resolution jointly from external and internal examples," *IEEE Trans. on Image Processing*, vol. 24, no. 11, pp.43594371, 2015.

[22] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," *Proc. of European Conf. Computer Vision (ECCV)*, pp. 184199, 2014.

[23] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295307, 2016.

[24] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to hand-written zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541551, 1989.

[25] J. Kim, J. Kwon Lee, and K. Mu Lee, "Accurate image super-resolution using very deep convolutional networks," *Proc. of IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 16461654, 2016.

[26] J. Kim, J. K. Lee, and K. M. Lee, "Deeply-recursive convolutional network for image super-resolution," *Proc. of IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 16371645, 2016.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proc. of IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 770778, 2016.

[28] C. Ledig, L. Theis, F. Huszàr, J. Caballero, A. Cunningham, A. Acosta, A. P. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," *Proc. of IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 4681-4690, 2017.

[29] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," *Proc. of IEEE Conf. Computer Vision and Pattern Recognition (CVPR) workshops*, vol. 1, no. 2, pp. 136-144, 2017.

[30] J. Engel, V. Usenko, and D. Cremers, "A photometrically calibrated benchmark for monocular visual odometry," arXiv:1607.02555, July 2016.

**Wonyeong Jeong** received his B.S. degree in Electrical and Computer Engineering from Seoul National University, Seoul, the Republic of Korea in August 2013, where he is currently working toward a Ph.D. degree with the Department of Electrical and Computer Engineering. His major research interests include SLAM, vision-based robotics applications, multi-agent system coordination, and semantic scene understanding.

**Jiyoun Moon** received her Bachelors of Science in Robotics from Kwangwoon University in August 2014. Her major research interests include Natural language process, Semantic scene understanding, and Mission planning.

**Beomhee Lee** received his B.S. and M.S. degrees in Electronics Engineering from Seoul National University, Seoul, Korea in 1978 and 1980, respectively, and a Ph.D. degree in Computer Information, and control engineering from the University of Michigan, Ann Arbor, MI, USA in 1985. He was an Assistant Professor with the School of Electrical Engineering at Purdue University, West Lafayette, IN from 1985 to 1987. He joined Seoul National University in 1987, and is currently a Professor with the Department of Electrical and Computer Engineering. His research interests include multi-agent system coordination, control, and application. Prof. Lee has been a Fellow of the Robotics and Automation Society since 2004.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.