Signal Processing Symposium

Ichiro TSUSHIN† Taro SHINGO‡

†Denshi University

‡Denshi Corp.

Abstract In this paper, we propose an example-based single image super resolution (SR) method by ℓ_2 approximation with self-sampled image patches. Example-based super resolution methods can reconstruct high resolution image patches by linear combination of atoms in overcomplete dictionary. This reconstruction requires a pair of two dictionaries created by tremendous low and high resolution image pairs from the prepared image databases. In our method, we introduce the dictionary by random sampling patches from only an input image without training. This dictionary exploits self-similarity of images and it will no more depend on external image set in regard to storage space or the accuracy of referred image set. In addition, we modified the approximation of input image to ℓ_2 norm minimization problem, instead of commonly used sparse approximation such as ℓ_1 norm regularization. The ℓ_2 approximation has an advantage of computational cost by only solving inverse problem. Through some experiments, the proposed method drastically reduces the computational time for SR, and provides comparable performance to conventional example-based SR methods with ℓ_1 approximation and dictionary training.

1 Example-based Super Resolution

Example-based SR algorithms deal with this problem by representing the LR image patch as combination of image patches and adding the regularizer to this combination coefficients [1], [2].

We describe one $\sqrt{n} \times \sqrt{n}$ patch of HR image as a vector replesented by $\boldsymbol{x} \in \mathbb{R}^n$. This patch $\boldsymbol{x}$ can be combinated by HR patch dictionary of K atoms $\boldsymbol{D}_h \in \mathbb{R}^{n \times K}$ and its coefficient vector $\boldsymbol{\alpha} \in \mathbb{R}^K$ shown as follow:

$$\boldsymbol{x} = \boldsymbol{D}_h \boldsymbol{\alpha} \tag{1}$$

The relation (1) is applied similary for representing the LR patch $\boldsymbol{y}$ which came from whole LR image $\boldsymbol{Y}$ using the LR patch dictionary $\boldsymbol{D}_l$ and its coefficient vector $\boldsymbol{\alpha}$:

$$\boldsymbol{y} = \boldsymbol{D}_l \boldsymbol{\alpha}. \tag{2}$$

Note that both two trained dictionaries $\boldsymbol{D}_h$ and $\boldsymbol{D}_l$ have the same representation $\boldsymbol{\alpha}$ for a certain image patch. In order to make this problem be well-posed, the sparse constraint is often imposed by ℓ_0 norm of $\boldsymbol{\alpha}$ shown as follows [3], [4]:

$$\boldsymbol{x} = \boldsymbol{D}_h \boldsymbol{\alpha} \qquad \text{with} \quad \|\boldsymbol{\alpha}\|_0 \ll K. \tag{3}$$

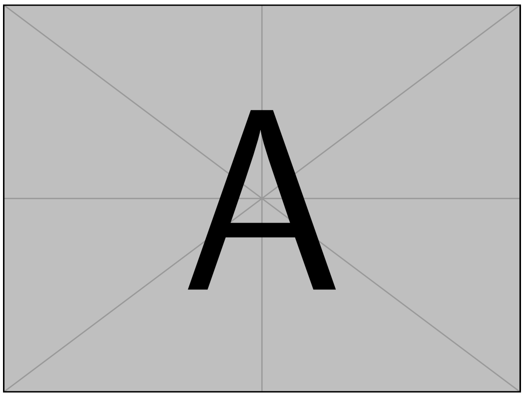


Fig. 1 Dictionary generation phase of proposed method

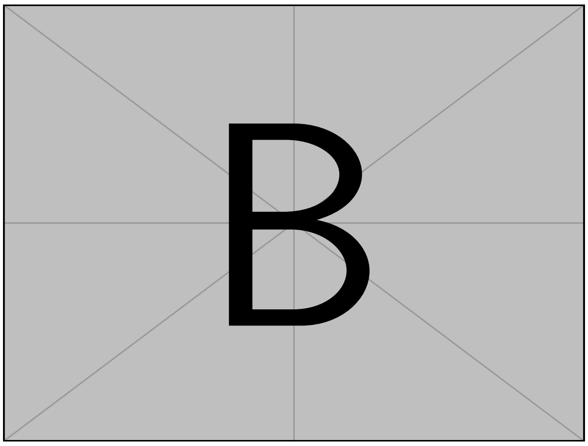


Fig. 2 SR reconstruction phase of proposed method

1.1 Self-sampled Dictionaries

In order to reconstruct the HR image $\boldsymbol{X}$, the pair of LR and HR dictionaries $\boldsymbol{D}_l, \boldsymbol{D}_h$ will be required. These dictionaries are usually generated from external training examples. In natural images, similar patches often appear not only in the image but also in its different scale images [5]. Based on this property, we produce LR and HR dictionaries from the input image $\boldsymbol{Y}$ alone.

1.2 HR patch reconstuction

With the optimal solution $\boldsymbol{\alpha}$ from (**??**), the HR patch feature can be estimated by $\hat{\boldsymbol{x}} = \boldsymbol{D}_h \alpha$. We modified the process to recover HR patch $\boldsymbol{x}$ from its feature $\hat{\boldsymbol{x}}$ from ℓ_1-based recovering refered in [6]. The procedure with ℓ_1 recovery is shown in Algorithm 1 and the modified procedure is in Algorithm 2. Each process reduces the DC component m of LR patch $\boldsymbol{y}$. This is because the

Algorithm 1 HR patch reconstruction process in [6]

for each LR patch $\boldsymbol{y} \in \mathbb{R}^{n\times 1}$ in $\boldsymbol{Y}$ **do**
  Set $m \leftarrow \text{mean}(\boldsymbol{y})$ as a DC component of $\boldsymbol{y}$ and,
  reduce the DC component from LR patch $\boldsymbol{y} \coloneqq \boldsymbol{y} - m\mathbf{1}$;
  Set $r \leftarrow \|\boldsymbol{y}\|_2$ as the norm of LR patch (before feature extraction;)
  Extract gradient feature $\hat{\boldsymbol{y}} \coloneqq \boldsymbol{F}\boldsymbol{y}$ for $\boldsymbol{y}$;
  Normalize the gradient feature $\hat{\boldsymbol{y}} \coloneqq \hat{\boldsymbol{y}}/\|\hat{\boldsymbol{y}}\|_2$;
  Estimate the dictionary coefficients $\boldsymbol{\alpha}$ (by ℓ_1 minimization, eq. **??**;)
  Recover HR patch feature $\hat{\boldsymbol{x}} = \boldsymbol{D}_x\boldsymbol{\alpha}$;
  Recover HR patch by adjusting its norm to LR patch's one:
  $\boldsymbol{x} \coloneqq (c \times r/\|\hat{\boldsymbol{x}}\|_2)\hat{\boldsymbol{x}} + m\mathbf{1}$ (c is emperically set constant);
  Add $\boldsymbol{x}$ to the corresponding pixels in HR image $\boldsymbol{X}$
end for

Algorithm 2 Modified HR patch reconstruction process

for each LR patch $\boldsymbol{y} \in \mathbb{R}^{n\times 1}$ in $\boldsymbol{Y}$ **do**
  Set $m \leftarrow \text{mean}(\boldsymbol{y})$ as a DC component of $\boldsymbol{y}$ and,
  reduce the DC component from LR patch $\boldsymbol{y} \coloneqq \boldsymbol{y} - m\mathbf{1}$;
  Extract gradient feature $\hat{\boldsymbol{y}} \coloneqq \boldsymbol{F}\boldsymbol{y}$ for $\boldsymbol{y}$;
  (Not normalizing the gradient feature $\hat{\boldsymbol{y}}$,) estimate the dictionary coefficients $\boldsymbol{\alpha}$ (by ℓ_2 minimization, eq. **??**;)
  Recover HR patch feature $\hat{\boldsymbol{x}} = \boldsymbol{D}_x\boldsymbol{\alpha}$;
  Recover HR patch by adding DC component $\boldsymbol{x} \coloneqq \hat{\boldsymbol{x}} + m\mathbf{1}$,
  (not scaling the dynamic range of HR patch feature;)
  Add $\boldsymbol{x}$ to the corresponding pixels in HR image $\boldsymbol{X}$
end for

feature extractive operation $\boldsymbol{F}$ cuts down the DC component of input patches $\boldsymbol{y}$, and the coefficient $\boldsymbol{\alpha}$ doesn't contain the information of the DC component in LR or HR patches.

2 Experimental results

In order to evaluate our image reconstruction framework, we conduct some experiments on 31 standard test images and apply the proposed SR method using ℓ_2 approximation. The size of all test images are 512×512, and we enlarged these images by factor 2 using SR algorithms, after shrinking manually by factor 2. In addition to our SR algorithm, general bicubic interpolation algorithm and ℓ_1 minimization SR algorithm refered in Section. 1 are evaluated for comparison. The differences between our algorithm and ℓ_1-based algorithm consist of two parts, ℓ_2-based minimization and self-sampled dictionary. Therefore we additionally evaluated the algorithms including one of our new-points, namely, algorithm using ℓ_2-based minimization with prepared dictionary and algorithm using ℓ_1-based minimization with self-sampled dictionary.

References

[1] S. Park, M. Park, and M. Kang, "Super-resolution image reconstruction: a technical overview," IEEE Signal Processing Magazine, vol.20, no.3, pp.21–36, 2003.

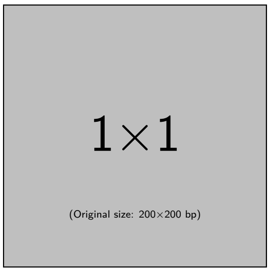
[2] H. Hou and H. Andrews, "Cubic splines for image interpolation and digital filtering," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol.26, no.6, pp.508–517, Dec. 1978.

[3] J. Yang, J. Wright, T. Huang, and Y. Ma, "Image super-resolution as sparse representation of raw image patches," Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, pp.1–8, June 2008.

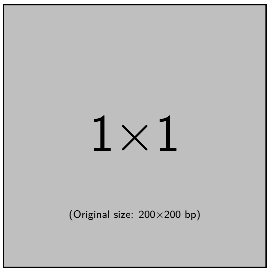
[4] J. Yang and J. Wright, "Image super-resolution via sparse representation," IEEE Transactions on Image Processing, vol.19, no.11, pp.2861–2873, May 2010.

[5] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," IEEE 12th International Conference on Computer Vision, pp.349–356, Sept. 2009.

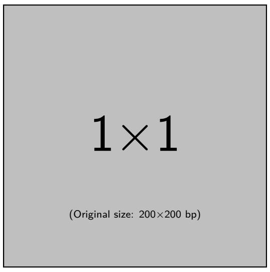
[6] J. Yang, Z. Wang, and Z. Lin, "Coupled Dictionary Training for Image Super-Resolution," IEEE Transactions on Image Processing, vol.21, no.8, pp.3467–3478, 2012.



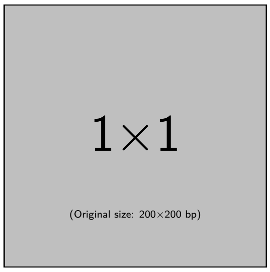
(a) Original 512 × 512 Image



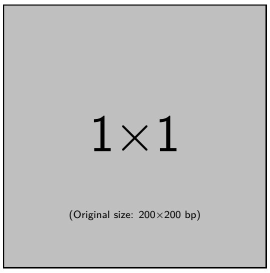
(b) Ground-truth HR



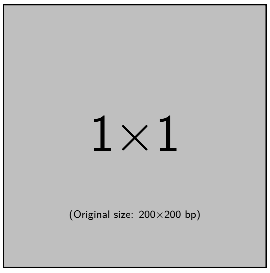
(c) Input LR Image



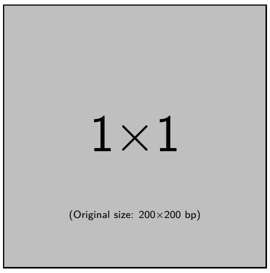
(d) Bicubic (PSNR = 30.869 dB)



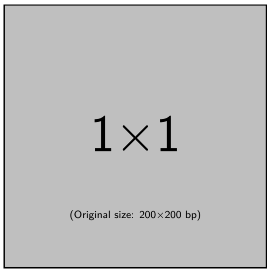
(e) Predefined dictionary (ℓ_1) (PSNR = 33.875 dB)



(f) Predefined dictionary (ℓ_2) (PSNR = 33.756 dB)



(g) Self-sampled dictionary (ℓ_1) (PSNR = 33.481 dB)



(h) Self-sampled dictionary (ℓ_2) (PSNR = 33.791 dB)

Fig. 3 Experimental result in Image "Airplane", compareing the letter part of size 64 × 64.

Table 1 Execution time compareing bicubic, ℓ_1, ℓ_2 algorithms with prepared dictionary and self-sampled dictionary. Execution time of each algorithm steps are also shown in example-based algorithms.

	Execution time (second)			
	Prepared		Self-sampled	
	L1	L2	L1	L2
Whole execution time	7.932	**0.541**	3.462	1.163
— with dictionary preparation	–	–	0.633	0.631
— with inverse matrix operation	–	0.026	–	0.025
— with optimization	7.494	0.073	2.390	0.073