Fast super-resolution using weighted median filtering

Andrey V. Nasonov, Andrey S. Krylov
Laboratory of Mathematical Methods of Image Processing
Faculty of Computational Mathematics and Cybernetics
Lomonosov Moscow State University
nasonov@cs.msu.ru , kryl@cs.msu.ru

***Abstract*—A non-iterative method of image super-resolution based on weighted median filtering with Gaussian weights is proposed. Visual tests and basic edges metrics were used to examine the method. It was shown that the weighted median filtering reduces the errors caused by inaccurate motion vectors.**

***Keywords*-super-resolution; weighted median filtering**

I. Introduction

Image resampling is one of the most important problems in image processing. Despite the increase of the resolution of modern camera sensors, this problem is still actual, for example, for old low-resolution video data and it is very important for surveillance applications.

Many image resampling algorithms use a priory information. As an example, image self-similarity at different resolutions is used in NEDI algorithm [1]. Nevertheless this approach improves the image visual quality only if the a priori information is true.

Super-resolution is an alternative method to obtain better results. Here several images of an object with subpixel shifts (see figure 1) are used to construct single high-resolution image [2], [3]. Camera sensors have non-zero size, and the observed pixel value is an approximation of image intensity in a certain area. If the object motion and the approximation function are known, then the information from all frames can be used to construct a single high-resolution image. The main problem of super-resolution is the requirement of accurate motion estimation. In this work, we propose a super-resolution method stable enough to the errors of motion vectors estimation.

II. Mathematical model

The super-resolution task can be posed as an inverse problem. The corresponding direct problem includes a set of the downsampling procedures. It produces the low-resolution images u_k after motion transformation and downscaling from the high-resolution image z as:

$$A_k z = u_k, \quad k = 1, 2, ..., N.$$

The operator A_k in the general case is represented as $A_k = DH_{cam}F_kH_{atm}z + n$ [4], where H_{atm} is the atmosphere blur, F_k is the motion operator, H_{cam} is the camera lens blur, D is a downscaling operator, n is a noise.

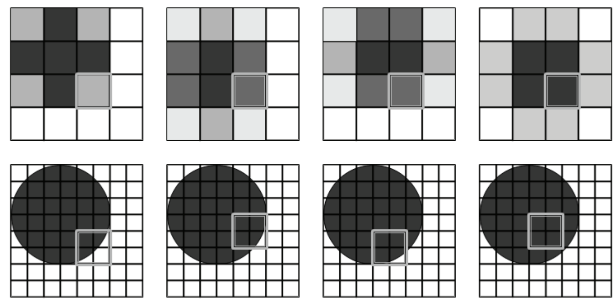


Figure 1. The correspondence between pixels of low-resolution images (top row) and pixels of high-resolution image (bottom row).

The atmosphere and camera lens blurs are modeled by a single Gauss filter H, and the system of equations takes the form

$$A_k z = DF_kHz = u_k, \quad k = 1, 2, ..., N. \qquad (1)$$

Motion estimation algorithms [5], [6] are used to calculate motion operators F_k. Many of them use optical flow method for accurate motion estimation [5]. Nevertheless the inaccuracy of the motion estimation remains very important in super-resolution and it is necessary to reduce its influence on the super-resolution result.

The super-resolution task described by the inverse problem for the system of equations (1) is an ill-posed problem. Iterative regularization methods based on Tikhonov regularization [7] are used to make this problem well-posed, but these methods are time consuming. Fast super-resolution uses non-iterative methods. For example, in [8] the authors calculate the average sum of upsampled and motion compensated low-resolution images $z_H = Hz$ and then sharpen the result using $z = H^{-1}z_H$ transformation. The idea of the proposed super-resolution method is close to this method but weighted median filtering with Gaussian weights is used.

III. Problem definition

We consider the super-resolution problem (1) with z and u_k defined on discrete sets $\{(i, j) : i, j \in \mathbb{Z}\}$. The motion transformation operator F_k defines a set of correspondences between the coordinates of points of source image and points of motion transformed image:

$$F_k z(i,j) = z(\widetilde{x}^k_{i,j}, \widetilde{y}^k_{i,j}).$$

Operator D performs downscaling:

$$Dz(x,y) = z(sx, sy),$$

where s is a downscaling factor, and the operator DF_k has the form

$$DF_k z(x,y) = z(\widetilde{x}^k_{si,sj}, \widetilde{y}^k_{si,sj}).$$

Using notation $(x^k_{i,j}, y^k_{i,j})$ for $(\widetilde{x}^k_{si,sj}, \widetilde{y}^k_{si,sj})$, the operator $A_k z = DF_k(Hz)$ can be written as

$$A_k z(i,j) = DF_k(Hz)(i,j) = (Hz)(x^k_{i,j}, y^k_{i,j}),$$

and the system of equations (1) takes the form

$$(Hz)(x^k_{i,j}, y^k_{i,j}) = u^k_{i,j}.$$

By omitting multi-dimensional indexing we compose several upsampled and motion transformed low-resolution images into a single image and rewrite $(x^k_{i,j}, y^k_{i,j}, u^k_{i,j})$ as (x_n, y_n, w_n). Finally we obtain the following super-resolution problem statement: to find the high-resolution image (Hz) with the known values w_n in the given points (x_n, y_n) (see figure 2):

$$(Hz)(x_n, y_n) = w_n. \qquad (2)$$

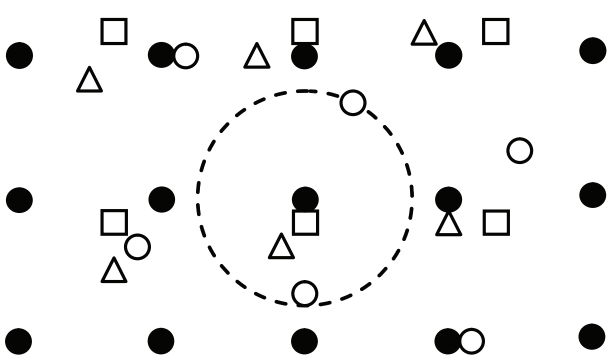


Figure 2. The illustration for super-resolution problem statement (2). Pixels of the low-resolution images are shown as □, ◯ and △. Pixels of the high-resolution image are shown as •.

IV. Problem solution

Even small errors in motion estimation result in serious degradation of the reconstructed image. If a super-resolution algorithm does not take into account the inaccuracy of motion vectors, it gives unsatisfactory results. Therefore instead of constructing the high-resolution image (Hz) which satisfies the equation (2) in every point (x_n, y_n), we use the following algorithm:

For every pixel (i,j) of the high-resolution image (Hz) we take all points (x_n, y_n) from a small neighborhood of (i,j) like shown in figure 2 and perform an averaging of these values.

A simple and fast averaging method is based on Gaussian filtering [8]:

$$(Hz)(i,j) = \frac{\sum_n w_n e^{-\frac{(x_n-i)^2+(y_n-j)^2}{2\sigma^2}}}{\sum_n e^{-\frac{(x_n-i)^2+(y_n-j)^2}{2\sigma^2}}}, \qquad (3)$$

where radius σ is chosen experimentally in accordance to the scale factor and accuracy of the motion vectors.

Gaussian averaging results in image blur. A robust approach based on median filtering was suggested in [9]. It upsamples the low-resolution images and combining the upsampled images using median averaging.

An adaptation of the median averaging to construct (Hz) is presented in [10]. It applies the median averaging to the values of all points in the neighborhood of the target pixel:

$$(Hz)(i,j) = \\ = \mathrm{med}(w_n : (x_n - i)^2 + (y_n - j)^2 < R^2). \qquad (4)$$

Median averaging produces sharp edges but it does not use the spatial distribution of points (x_n, y_n). To take the benefits of both Gauss averaging (3) and median averaging (4), we propose a combined method based on weighted median averaging. In weighted median $\mathrm{wmed}(w_n, c_n)$, every value w_n has a weight c_n. We choose the weights c_n as in the Gauss averaging:

$$c_n = \exp\left(-\frac{(x_n - i)^2 + (y_n - j)^2}{2\sigma^2}\right).$$

If the weights c_n are natural numbers, we calculate the weighted median as median average with w_n taken c_n times. To find the result of the median averaging $\mathrm{wmed}(w_n, c_n)$ in the general case, the pairs (w_n, c_n) are sorted in the ascending order of w_n. Next, we find the value m which satisfies the conditions

$$\sum_{k=1}^{m-1} c_k \le S/2, \quad \sum_{k=1}^{m} c_k > S/2, \quad S = \sum_k c_k,$$

and take the value w_m as the result of the weighted median averaging.

This process is illustrated in figure 3. We represent every pair (w_n, c_n) as a rectangle with the width c_n and fixed height. Next we construct a long rectangle with width S by connecting these rectangles in the ascending order of w_n. Finally we take the value w_m of the rectangle in the middle of the constructed rectangle as the result of the weighted median.

The obtained result is the approximation of the blurred image (Hz). But actually the proposed method produces sharp images. This is caused by the behavior of the median filtering. If we increase the radius of the median filtering then

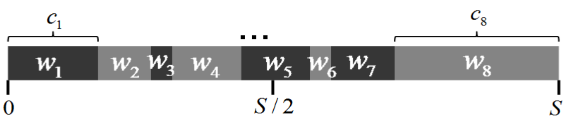


Figure 3. Weighted median averaging procedure.

we do not increase the image blur of the resulting image and the resulting image tends to be piece-wise flat. Therefore we not apply deblur algorithms to the result and take the result of the weighted median averaging as the result of super-resolution.

V. Results

To test the results of the proposed method we applied random shifts to reference high-resolution images, and the obtained images were downsampled. Then the high-resolution images were reconstructed from these low-resolution images by the super-resolution algorithm and compared with the reference images. To test the stability of the proposed method to the motion vectors errors, we added noise to the motion vectors. A random uniformly distributed value in $[-1, 1]$ range was added to every motion vector for a quarter of the input images. A random value in $[-0.25, 0.25]$ range was added to motion vectors for other images.

The results for the proposed super-resolution method are shown in figure 4. It can be seen that the proposed method has better visual quality than of the method based on median averaging and the method based on Gaussian averaging. Median averaging produces noisy edges while Gaussian averaging blurs the edges in the case of errors in motion vectors estimation.

To measure numerically the quality of the result of the super-resolution algorithm, we used edge adaptive metrics from [11]. The metrics BEP (Basic Edges Points) calculates the mean square error (MSE) in the points of sharp and isolated edges — basic edges. The metrics BEN (Basic Edges Neighborhood) calculates the MSE in the basic edges neighborhood.

The weighted median averaging shows better BEN and BEP than the Gaussian averaging, but the results are contradictory in comparison to the median averaging. The weighted median averaging gives better results than the median averaging in edge areas (BEP areas), but is worse in the areas near the edges (BEN areas). Nevertheless the problem of edge reconstruction is usually more significant than the reconstruction of non-edge areas in the case of erroneous motion vectors. Thus we make a conclusion that the weighted median averaging method shows better results than the median averaging method or the Gaussian averaging method.

The results can be improved by varying the Gauss radius σ and the radius R for different pixels of the image. This will be part of the future work.

VI. Conclusion

Non-iterative image super-resolution method based on the weighted median averaging has been proposed. It was shown that weighted median averaging reduces the errors caused by inaccurate motion vectors. The work was supported by RFBR grant 10-01-00535 and by grant of Human Capital Foundation.

References

[1] X. Li and M. Orchard, "New edge-directed interpolation," *IEEE transactions on image processing*, vol. 10, no. 10, pp. 1521–1527, 2001.

[2] S. Borman and R. Stevenson, "Super-resolution from image sequences - a review," *Midwest Symposium on Circuits and Systems*, pp. 374–378, 1998.

[3] K. Malczewski and R. Stasinski, "Super resolution for multimedia, image, and video processing applications," *Recent Advances in Multimedia Signal Processing and Communications*, vol. 231, pp. 171–208, 2009.

[4] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar, "Fast and robust multi-frame super-resolution," *IEEE Transactions on Image Processing*, vol. 13, no. 10, pp. 1327–1344, 2004.

[5] A. Bruhn, J. Weickert, and C. Shnorr, "Lucas/Kanade Meets Horn/Schunck: Combining Local and Global Optic Flow Methods," *International Journal on Computer Vision*, vol. 61, no. 3, pp. 211–231, 2005.

[6] H. Le and G. Seetharaman, "A super-resolution imaging method based on dense subpixel-accurate motion fields," *Proc. of the Third International Workshop on Digital and Computational Video*, pp. 35–42, 2002.

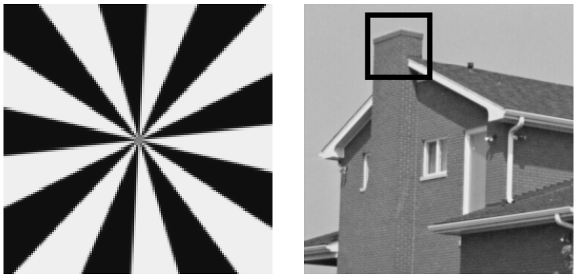
[7] A. N. Tikhonov and V. Y. Arsenin, *Solutions of Ill-Posed Problems*. Washington DC: WH Winston, 1977.

[8] A. S. Krylov, A. V. Nasonov, and O. S. Ushmaev, "Video super-resolution with fast deconvolution," *Pattern Recognition and Image Analysis*, vol. 19, no. 3, pp. 497–500, 2009.

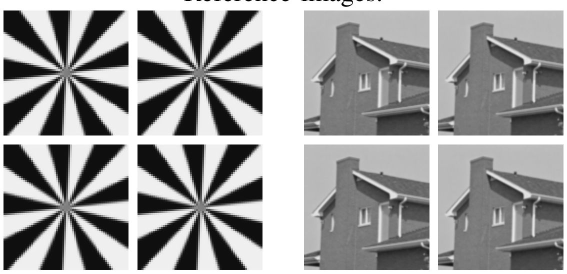
[9] A. Zomet, A. Rav-acha, and S. Peleg, "Robust super-resolution," *Proc. of the IEEE Workshop on Applications of Computer Vision*, pp. 645–650, 2001.

[10] A. Sanchez-Beato and G. Pajares, "Robust super-resolution using a median filter for irregular samples," *Lect. Notes in Comp. Science*, vol. 5524, pp. 298–305, 2009.

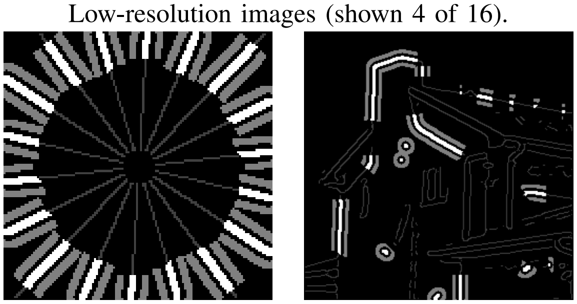
[11] A. Nasonov and A. Krylov, "Adaptive image deringing," *Proceedings of GraphiCon'2009*, pp. 151–154, 2009.



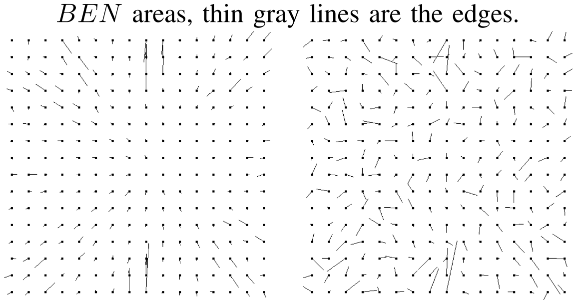
Reference images.



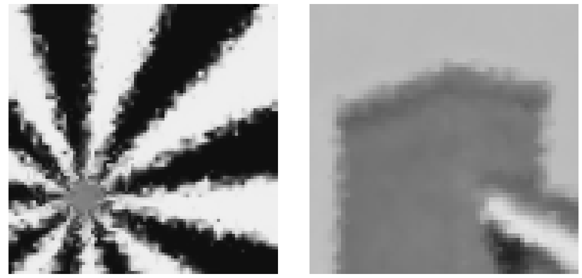
Low-resolution images (shown 4 of 16).



BEP and BEN areas.
White areas are BEP areas, gray areas are BEN areas, thin gray lines are the edges.

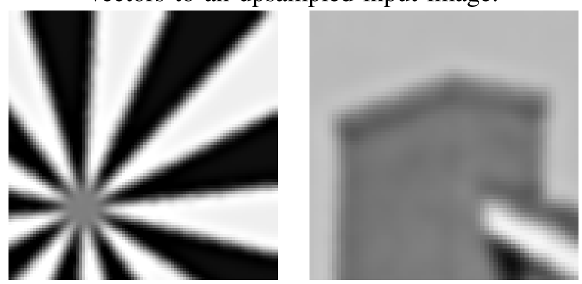


Example of estimated motion vectors (left image) and corresponding erroneous motion vectors (right image).



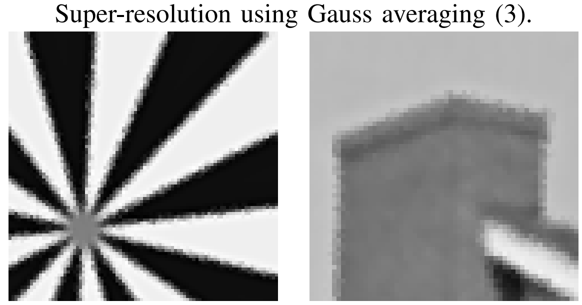
$BEP = 55.34$, $BEN = 0.26$ | $BEP = 17.00$, $BEN = 5.36$

The result of applying erroneous motion vectors to an upsampled input image.



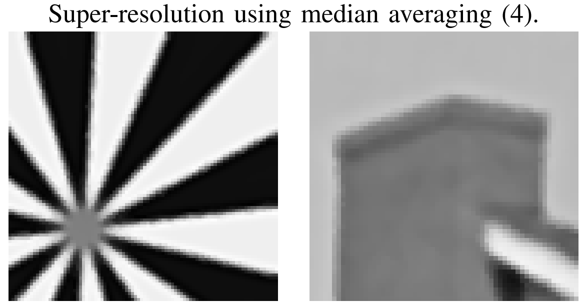
$BEP = 35.0$, $BEN = 5.63$ | $BEP = 9.73$, $BEN = 4.47$

Super-resolution using Gauss averaging (3).



$BEP = 25.73$, $BEN = 0.22$ | $BEP = 9.08$, $BEN = 3.66$

Super-resolution using median averaging (4).



$BEP = 20.09$, $BEN = 0.30$ | $BEP = 6.89$, $BEN = 3.94$

Super-resolution using weighted median averaging.

Figure 4. The illustration of super-resolution methods for scale factor 2 and 16 initial low-resolution images. Parameter $\sigma = 1.2$.