

# Edge-Directed Interpolation in a Bayesian Framework

Karen Simonyan  
simonyan@graphics.cs.msu.ru

Graphics & Media Lab  
Lomonosov Moscow State University  
Moscow, Russia

Dmitriy Vatolin  
dmitriy@yuvsoft.com

YUVsoft Corp.  
Moscow, Russia

---

## Abstract

In this paper we present a novel framework for Edge-Directed Interpolation (EDI) of still images. The problem is treated as finding maximum a posteriori estimates of each interpolated pixel type and intensity value. The pixel type may be one of the pre-defined edge directions or "non-edge". Instead of the separate steps of edge orientation detection and intensity interpolation, maximizing the joint probability density function of type and intensity provides a better fit to the local image structure. Such a technique allows an effective discrimination between edges and non-edges (uniform areas and texture), thus leading to the suppression of artifacts which are common to existing EDI methods. Objective and subjective comparisons with conventional EDI methods corroborate the advantages of the proposed one. Moreover, the locality and the low computational complexity of the method make it suitable for a hardware implementation.

## 1 Introduction

The problem of image interpolation is one of the most thoroughly developed in the area of multimedia processing. The reason is that image upscaling is an important part of many image and video processing algorithms, *e.g.* super-resolution [1], and a distinct task as well. A special attention has been paid to the problem since the emergence of High Definition Television (HDTV). Since most video content is still lower resolution, there is a strong demand of fast high-quality upscaling algorithms. Linear interpolation methods suffer from aliasing and Hibbs effect, and they fail to provide artifacts-free enlargement of edges. This has led to a plethora of edge-oriented upscaling techniques aimed at providing superior edge processing quality. However, edge-directed processing of textured areas results in strong artifacts, making the discrimination between edges and texture an important task as well.

In this paper we propose a Bayesian framework for Edge-Directed Interpolation (EDI) which allows a simultaneous search for pixel type and intensity which maximize the joint Probability Density Function (PDF). The pixel type denotes if the pixel lies on an edge of a certain direction or on a non-edge. The search is performed independently for each pixel of a High Resolution (HR) frame, which means that the method is local and suits well for a hardware implementation. As objective and subjective comparisons show, the algorithm

outperforms conventional EDI approaches, providing not only high-quality edges upscaling, but artifacts-free texture interpolation as well.

The rest of the paper is organized as follows. In Section 2 a review of the related work is given. In Section 3 the proposed algorithm is described. Experimental results are presented in Section 4. Section 5 concludes the paper and points possible directions of the future research.

## 2 Related work

Non-linear image enlargement techniques can be divided into two groups according to the edge processing strategy which can be implicit or explicit.

### 2.1 Implicit edge processing

Such methods do not perform explicit edge detection; instead of that, inner parameters adaptively change depending on the local image structure. Therefore, the interpolation is done along an edge if it is present. One of the possible approaches to the implicit edge processing is making a linear interpolation method (*e.g.* bilinear or bicubic) dependent on the local intensity variations. Ramponi [7] proposed to use the warped distance to the interpolated pixel instead of a regular one, thus modifying the one-dimensional kernel of a separable interpolation filter. The computational complexity of the method remains low, but the upscaling quality of edges oriented neither horizontally nor vertically is poor. A certain improvement was brought by Hwang and Lee [8], who suggested to modify the two-dimensional interpolation kernel by introducing the local gradient features. However, the necessity to adjust a sharpness parameter for each image and the lack of a special textured area processing form the drawbacks of the algorithm. Another approach, NEDI, was proposed by Li and Orchard in [9]. The optimal interpolation in terms of the mean squared error was suggested under the assumption of the local intensity covariance being constant for both Low Resolution (LR) and HR images. The method succeeds in upscaling edges of different orientations, but in the areas of texture and clutter the assumption made becomes false, leading to unnatural images and even strong artifacts. Moreover, the computational complexity of NEDI makes it inappropriate for the real-time processing. Recently the mathematical apparatus of kernel regression was applied by Takeda *et al.* [10] to various image processing problems including interpolation. The usage of a special steering kernel which adjusts its parameters to edge orientation and strength allowed high-quality edges upscaling. Nevertheless, from our experiments (see Section 4) the resulting HR image in whole is overblurred; furthermore, artifacts appear in some cases. Again, the amount of computations required makes it difficult to adapt the method for fast processing.

Unlike local methods mentioned above, global upscaling methods impose constraints on the whole resulting image. Schultz and Stevenson [8] used Bayesian reasoning with discontinuity-preserving prior for finding Maximum A Posteriori (MAP) estimate of HR image given an LR one. In [2] Freeman *et al.* proposed an example-based image magnification method. They argued that the relationship between medium- and high-frequency image patches can be exploited to add high-frequency details to the enlarged image. A special database was introduced to store patch correspondences. To stitch the patches together a Markov Random Field (MRF) was used. The algorithm performance crucially depends on a

database used: it must be large enough to ensure proper handling of different LR input. Both global methods mentioned above exhibit enormous computational complexity.

## 2.2 Explicit edge processing

An algorithm belonging to this group can be subdivided into two blocks: edge detection and edge-oriented interpolation. Such methods often limit the set of possible edge directions, aiming at speed-up and robustness. Muresan [6] suggested to consider only four possible edge directions:  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ , and  $135^\circ$  relative to the horizontal axe. The proper direction is determined via the analysis of pixel intensity differences in the possible edge directions. Linear interpolation is then performed along the detected edge orientation. Though fast, the algorithm performs poorly in the case of edges not aligned with the aforementioned directions. Additionally, specific curly artifacts appear in the textured areas. A more flexible method was proposed by Wang and Ward in [10]. At first, the intensity gradient values are calculated for HR pixels via the bilinear interpolation from LR pixel gradients. Secondly, interpolation is done using a 4-point kernel oriented in the edge direction (orthogonal to the HR pixel gradient). The set of possible edges is limited to 20 directions between  $0^\circ$  and  $180^\circ$ . A special processing is also held for ridge pixels and edges with a high curvature. The main drawback of the method is the instability of an HR pixel gradient calculation: in the cluttered areas the bilinear interpolation produces erroneous results. In [9] Li and Nguyen introduced an MRF with cliques aligned with 16 pre-defined possible edge directions. Edge consistency in the upscaled image was ensured via special potential functions which take into account both edge strength and interpolated pixel intensity. A special processing was suggested to detect non-edge pixels. To increase the processing speed, a small candidate set was introduced for each HR pixel, and a single-pass suboptimal search procedure was offered. According to our experiments (see Section 4), the main drawback of the approach is that minimizing the sum of potentials we get an intensity value that to a certain degree fits all the directions at the same time. This leads to specific aliasing effect near some edges.

## 3 Proposed EDI algorithm

### 3.1 The proposed framework: a general case

We suggest applying a pixelwise Bayesian reasoning for the *simultaneous* HR pixel type detection and its intensity interpolation. The pixel type can be either an index of one of the pre-defined edge orientations or "non-edge". The former is for the pixels lying on edges; the number of allowed orientations in the range  $[0^\circ; 180^\circ]$  is denoted by  $D$  (see Fig. 1). The latter stands for the pixels belonging to uniform areas, areas of texture and clutter. The usage of the per-pixel optimization, not global one, was driven by the aim to create a local framework which suits well for a hardware implementation. In the general case we introduce an iterative upscaling process, at each iteration  $n$  considering MAP estimates of each HR pixel  $j$  type  $t^n(j)$  and intensity  $I^n(j)$  given types and intensities calculated at the previous iteration  $(n - 1)$ :

$$\left\{ \tilde{t}^n(j), \tilde{I}^n(j) \right\} = \arg \max_{t^n(j), I^n(j)} P(t^n(j), I^n(j) | \{t^{n-1}(k), I^{n-1}(k)\}_{k \in \Theta}) \quad \forall j. \quad (1)$$

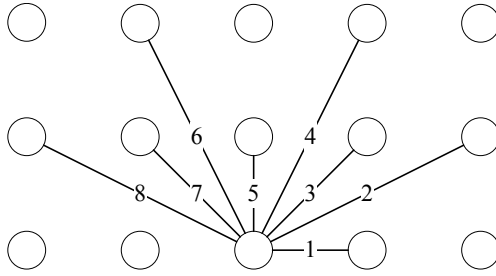


Figure 1: Possible edge orientations for  $D = 8$  and their indexes. LR pixels are shown by circles.

Here  $\Theta$  stands for a spatial neighbourhood of pixel  $j$ . According to the Bayes rule and omitting obvious transformations, MAP estimates (1) can be rewritten as:

$$\begin{aligned} \{\tilde{t}^n(j), \tilde{I}^n(j)\} &= \arg \max_{t^n(j), I^n(j)} (P(I^n(j) | t^n(j), \{I^{n-1}(k)\}_{k \in \Theta}) \times \\ &\times P(t^n(j) | \{t^{n-1}(k), I^{n-1}(k)\}_{k \in \Theta})) = \arg \max_{t^n(j), I^n(j)} (P(I^n(j) | t^n(j), \{I^{n-1}(k)\}_{k \in \Theta}) \times (2) \\ &\times P(\{I^{n-1}(k)\}_{k \in \Theta} | t^n(j)) \times P(t^n(j) | \{t^{n-1}(k)\}_{k \in \Theta})). \end{aligned}$$

As it can be noticed, the problem took the form of maximizing the product (2) of three PDFs which must be specified in some way. The first one,  $P_1 = P(I^n(j) | t^n(j), \{I^{n-1}(k)\}_{k \in \Theta})$ , denotes the likelihood of an intensity value  $I^n(j)$  given the pixel type  $t^n(j)$  and neighbourhood intensities  $\{I^{n-1}(k)\}_{k \in \Theta}$  calculated at the previous iteration. In other words,  $P_1$  reflects how good the new pixel intensity fits the local image structure while the new pixel type is being fixed. The second PDF,  $P_2 = P(\{I^{n-1}(k)\}_{k \in \Theta} | t^n(j))$ , expresses how likely the intensities  $\{I^{n-1}(k)\}_{k \in \Theta}$  are if the  $j$ -th pixel has type  $t^n(j)$ . Thus  $P_2$  measures how the pixel type impacts its neighbourhood intensities. The last component,  $P_3 = P(t^n(j) | \{t^{n-1}(k)\}_{k \in \Theta})$ , expresses the relation between the new pixel type  $t^n(j)$  and the old neighbourhood configuration  $\{t^{n-1}(k)\}_{k \in \Theta}$ . In a general case the optimum search is performed over all possible pairs  $\{t^n(j), I^n(j)\}$ .

### 3.2 Framework implementation

Here we describe one of the possible implementations of the framework described in Subsection 3.1. The major assumptions made are the following:

- The intensities of pixels lying next to an edge are distributed as a mixture of gaussians with means corresponding to different sides of the edge.
- The intensities of non-edge pixels are uniformly distributed as no specific information is available for them. Therefore, their PDF equals to  $\frac{1}{256}$  if the 8-bit intensity coding is used.

According to this, the following derivations are appropriate for  $P_1 - P_3$ :

$$P_1 = \begin{cases} \frac{1}{256}, & \text{if } t^n(j) = \text{"non-edge"}; \\ \frac{1}{\sqrt{2\pi}\sigma_1} \exp\left(-\frac{(I^n(j)-\mu)^2}{2\sigma_1^2}\right), & \text{otherwise,} \end{cases} \quad (3)$$

where  $\mu$  is the mean of  $2N_1 + 1$  pixels of  $\{I^{n-1}(k)\}_{k \in \Theta}$  centered at the current pixel  $j$  and lying on an edge with direction  $t^n(j)$ . In other words, denoting the coordinates of the current pixel by  $(x, y)$ , we get

$$\mu = \frac{1}{2N_1 + 1} \sum_{p=-N_1}^{N_1} I^{n-1}(x + p\Delta X(t^n(j)), y + p\Delta Y(t^n(j))), \quad (4)$$

where  $(\Delta X(t), \Delta Y(t))$  is the distance vector between two neighboring HR pixels lying on an edge with direction  $t$ , e.g.  $(\Delta X(4), \Delta Y(4)) = (1, -2)$  for  $D = 8$ . Deviation  $\sigma_1$  is assumed constant.

The value of  $P_2$  depends on whether the current pixel belongs to a non-edge or an edge:

$$P_2 = \begin{cases} \left(\frac{1}{256}\right)^{|\Theta|}, & \text{if } t^n(j) = \text{"non-edge"}; \\ \prod_{i=0}^2 \left( \prod_{u \in E(i)} \frac{1}{\sqrt{2\pi}\sigma_2} \exp\left(-\frac{(I^{n-1}(u) - \mu_i)^2}{2\sigma_2^2}\right) \right) \cdot \left(\frac{1}{256}\right)^{|\Theta| - \sum_{i=0}^2 |E_i|}, & \text{otherwise.} \end{cases} \quad (5)$$

In the first case, if the pixel does not belong to an edge, the distribution of all  $|\Theta|$  pixel values in its neighbourhood is considered uniform. In the other case, if the pixel belongs to an edge of  $t^n(j)$  direction, three groups  $\{E_i\}_{i=0}^2$  of pixels oriented in the direction  $t^n(j)$  are taken into account as illustrated in Fig. 2. Group  $E_0$  contains  $2N_2 + 1$  edge pixels including the current one at which the group is centered. Each of the groups  $E_1$  and  $E_2$  is formed by  $2N_3$  pixels adjacent to the edge.

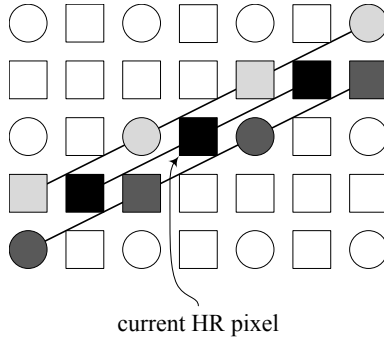


Figure 2: Spatial neighbourhood  $\Omega$  of the current HR pixel  $j$  for the case  $D = 8$ , 2 times magnification. LR pixels are shown by circles, interpolated pixels – by squares. Pixels lying next to an edge of index  $t^n(j) = 2$  are shown with gradations of black: pixels from  $E_0$  are painted with black, from  $E_1$  and  $E_2$  – with light and dark grey (here  $N_2 = 1$ ,  $N_3 = 2$ ). The intensities of pixels painted with white are considered uniformly distributed.

The intensities in all groups are assumed normally distributed with means  $\{\mu_i\}_{i=0}^2$  and constant deviation  $\sigma_2$ . The means are computed similar to (4):

$$\begin{aligned} \mu_0 &= \frac{1}{2N_2 + 1} \sum_{p=-N_2}^{N_2} I^{n-1}(x + p\Delta X(t^n(j)), y + p\Delta Y(t^n(j))); \\ \mu_{\{1,2\}} &= \frac{1}{2N_3} \sum_{p=-N_3+1}^{N_3} I^{n-1}(\tilde{x}_{\{1,2\}} + p\Delta X(t^n(j)), \tilde{y}_{\{1,2\}} + p\Delta Y(t^n(j))). \end{aligned} \quad (6)$$

Here  $(\tilde{x}_{\{1,2\}}, \tilde{y}_{\{1,2\}})$  stands for the pixels from  $E_{\{1,2\}}$  lying next to the current pixel  $(x, y)$  (on the left and on the right in Fig. 2). And finally, pixels from  $\Theta$  not belonging to  $\cup_{i=0}^2 E_i$  are assumed having a uniform distribution.

To estimate  $P_3$  we employ the heuristic that if the type of a pixel is an edge of a certain direction, then the neighboring pixels in the same or close direction usually exhibit the same or close edge orientations. Non-edge pixels are also likely to form consistent groups in the image plane. This leads to the following equation:

$$P_3 = \frac{1}{Z} \exp - \sum_d \left( w(t^n(x, y), d) \cdot \sum_{s \in \{-1, 1\}} \delta(t^n(x, y), t^{n-1}(x + s\Delta X(d), y + s\Delta Y(d))) \right). \quad (7)$$

Here  $\delta(t_1, t_2)$  is a distance function between the pixel types  $t_1$  and  $t_2$ :

$$\delta(t_1, t_2) = \begin{cases} 0, & \text{if both } t_1 \text{ and } t_2 \text{ are "non-edge";} \\ V, & \text{if only one of } t_1 \text{ and } t_2 \text{ is "non-edge";} \\ (\min(|t_1 - t_2|, D - |t_1 - t_2|))^\gamma, & \text{otherwise.} \end{cases} \quad (8)$$

In (8)  $V$  is a constant specifying the distance between edge and non-edge types and  $\gamma$  is a parameter which controls the influence of neighbour pixels type on the current pixel type. From (8) follows that if both  $t_1$  and  $t_2$  are edge types, the distance between them depends on how close the corresponding orientations are, reaching maximum if they are orthogonal.  $w(t_1, t_2)$  is a weighting function which takes the maximum value when the types  $t_1$  and  $t_2$  are equal:

$$w(t_1, t_2) = \exp(-\delta(t_1, t_2)). \quad (9)$$

Finally,  $Z$  is a normalizing constant ensuring  $\sum_{t^n(j)} P_3 = 1$ . In (7) the summation is performed over all possible edge directions  $d$ , and for each of them it is determined how much the types of pixels  $(x \pm \Delta X(d), y \pm \Delta Y(d))$  are close to type  $t^n(x, y)$ . The weighting allows to account directions  $d$  close to  $t^n(x, y)$  with higher weights, thus enforcing the edge consistency.

Initial estimates of HR pixels intensities and types have a crucial effect on the algorithm performance. The intensity values were obtained via the bicubic interpolation filter. As for the initial pixel types, the only function depending on it is  $P_3^1 = P(t^1(j) \{t^0(k)\}_{k \in \Theta})$  and from our experiments better results can be achieved specifying not  $t^0$  itself but  $P_3^1$  as:

$$P_3^1 = \begin{cases} 0, & \text{if } t^1(j) = \text{"non-edge"}; \\ 1/D, & \text{otherwise.} \end{cases} \quad (10)$$

*I.e.* at the 1st iteration only edge types are considered possible (with a uniform distribution), and non-edge areas detection is done on further iterations.

Substituting (3), (5), and (7) into (2) we get:

$$\begin{aligned} \{ \tilde{t}^n(j), \tilde{I}^n(j) \} &= \arg \max_{t^n(j), I^n(j)} (P_1 \cdot P_2 \cdot P_3) = \arg \min_{t^n(j), I^n(j)} -\ln(P_1 \cdot P_2 \cdot P_3) = \\ &= \arg \min_{t^n(j), I^n(j)} (L_{12} + L_3), \end{aligned} \quad (11)$$

where (omitting obvious transformations)

$$L_{12} = \begin{cases} \mathcal{L}, & \text{if } t^n(j) = \text{"non-edge"}; \\ \alpha(I^n(j) - \mu)^2 + \sum_{i=0}^2 \sum_{u \in E(i)} (I^{n-1}(u) - \mu_i)^2, & \text{otherwise,} \end{cases} \quad (12)$$

$$L_3 = \beta \sum_d \left( w(t^n(x,y), d) \cdot \sum_{s \in \{-1,1\}} \delta(t^n(x,y), t^{n-1}(x + s\Delta X(d), y + s\Delta Y(d))) \right). \quad (13)$$

For simplicity, in (12) – (13) we introduced several new constants:

$$\mathcal{L} = 2\sigma_2^2 \left( \ln \frac{256}{\sqrt{2\pi}\sigma_2} \sum_{i=0}^2 |E_i| + \ln \frac{256}{\sqrt{2\pi}\sigma_1} \right), \alpha = \left( \frac{\sigma_2}{\sigma_1} \right)^2, \beta = 2\sigma_2^2. \quad (14)$$

It is worth noting that according to (11), the proposed EDI method can also be treated as the energy minimization problem. Like in [9], to lower the computational complexity of the minimum search it is performed over a reduced candidate set

$$\Omega = \bigcup_{t \in \{1..D\}} \{ \{t, I^{\text{EDI}}(t)\}; \{t, I^{\text{Bic}}\} \} \cup \{ \{ \text{"non-edge"}, I^{\text{Bic}} \} \} \quad (15)$$

As it can be noticed, for each edge type  $t \in \{1..D\}$  only two possible intensity values are considered:

- EDI candidate  $I^{\text{EDI}}(t)$  which is calculated in the bilinear manner using LR pixels forming a parallelogram aligned with the edge direction  $t$ ; a similar approach to the edge-oriented interpolation was taken in [10]. *E.g.* in Fig. 3 the parallelogram consists of pixels marked E, F, G, H and the EDI candidate is calculated as the mean of corresponding intensities.
- Bicubic candidate  $I^{\text{Bic}}$  which is taken directly from the HR frame obtained via the bicubic interpolation.

Only the bicubic candidate is considered for the "non-edge" pixel type. This prevents applying edge-directed interpolation to textured areas, which results in annoying artifacts.

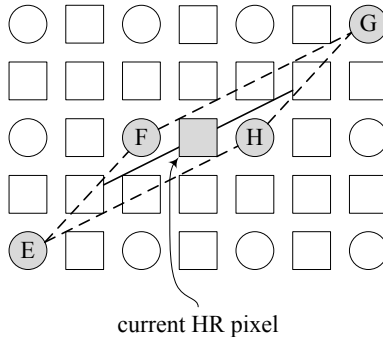


Figure 3: Calculation of EDI candidate  $I^{\text{EDI}}(t)$  for the case  $D = 8$ ,  $t = 2$ , 2 times magnification. LR pixels are shown by circles, interpolated pixels – by squares. The interpolation parallelogram is shown by a dashed line.

From (11) – (13), (15) follows that the proposed EDI method can be implemented using integer arithmetic without divisions. The locality of the approach (to process the current HR pixel only its neighbourhood  $\Theta$  must be known) makes the algorithm parallelization possible.

## 4 Experimental results

The goal of the comparison was to test the upscaling algorithms performance on an input data well approximating the real-world cases where different types of *both* edges and texture are present. We used 3 ground-truth greyscale (luminosity only) frames of  $1280 \times 704$  videos *Jets*, *Panslow*, and *Spincalendar*. Direct downsampling of the HR images leads to the aliased picture which is uncommon in real applications. Therefore, the ground-truth was at first low-pass filtered by a symmetrical  $3 \times 3$  Gaussian kernel with  $\sigma = 1$ . Then the blurred HR images were downsampled 2 times to  $640 \times 352$  LR images which were used as an input to the upscaling methods. After the 2 times enlargement the resulting HR images were obtained. As the input LR images were derived from blurred HR images, the result of the interpolation was also blurred. Because of that, a direct comparison of the resulting pictures with the ground-truth was quite incorrect, and deblurring was introduced as an additional postprocessing step. The deblurred resulting HR images were then compared with the ground-truth using the PSNR measure. It is worth noting that such a comparison scheme is often used in super-resolution algorithms tests [10, 11]. We used MATLAB Image Processing Toolbox implementation of the Richardson-Lucy Deconvolution (RLD) algorithm, which was run with the aforementioned Gaussian convolution kernel, and RLD deblurring iterations number was varied from 1 to 10. Providing the deblurring algorithm with a known blur kernel is plausible here as the performance of interpolation is tested, not that of deblurring.

We compared the proposed EDI algorithm with a conventional bicubic interpolation and EDI algorithms which implementations were provided by their authors: MRF-EDI [12], Steering Kernel Regression (SKR) [13], and NEDI [6]. The algorithms were executed with default parameter values, and for the proposed method the following parameter set was used:  $D = 8$ ,  $N_1 = 1$ ,  $N_2 = 1$ ,  $N_3 = 2$ ,  $\sigma_1^2 = 32$ ,  $\sigma_2^2 = 256$ ,  $V = 2$ ,  $\gamma = 2$ , 2 processing iterations.

The dependence of PSNR on RLD iterations number for the compared methods is shown in Fig. 4 – Fig. 6 (0 iterations stand for the case where no RLD deblurring was used and the blurred HR output of the methods was directly compared with the ground-truth). From the plots it is clear that the proposed method outperforms other EDI methods and the bicubic interpolation as well. Good results of bicubic upscaling comparing to some EDI techniques are explained by a large amount of fine texture in the test images, where artifacts and overblurriness inherent to other EDI algorithms lead to the PSNR degradation. The proposed method efficiently discriminates between edges and texture, gaining the best PSNR results.

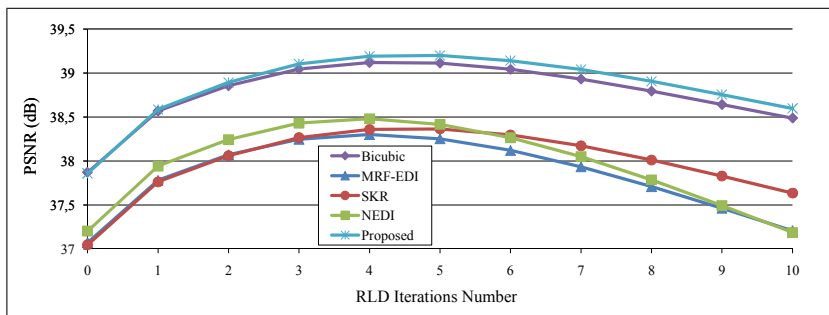


Figure 4: Dependence of PSNR on RLD iterations number at *Jets* test image.

Visual performance of the proposed algorithm is presented in Fig. 7 – Fig. 8 along with the results of other methods. For clarity, images in Fig. 7 are enlarged two times by the



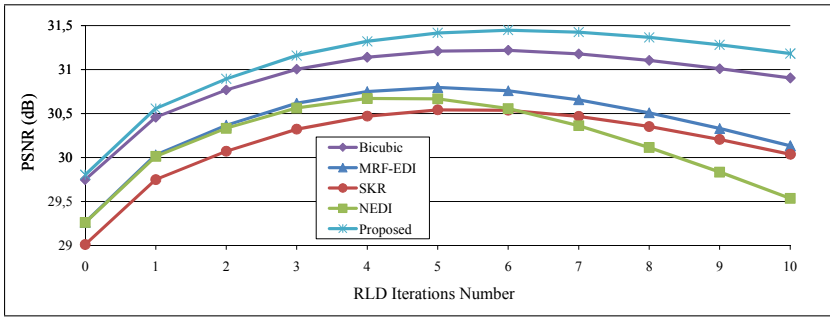


Figure 5: Dependence of PSNR on RLD iterations number at *Panslow* test image.

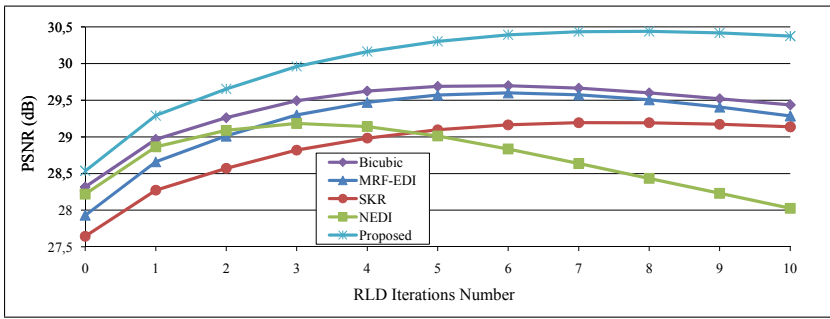


Figure 6: Dependence of PSNR on RLD iterations number at *Spincalendar* test image.

nearest neighbor filter. In both figures images are shown without deblurring. As it can be seen, our method provides better edges upscaling comparing to the bicubic interpolation and MRF-EDI, more detailed and sharp picture comparing to SKR, and the resulting picture is not contaminated with artifacts as opposed to NEDI.

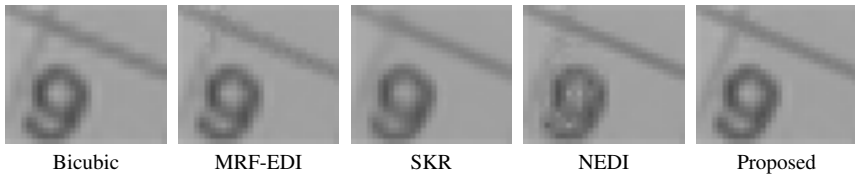


Figure 7: Visual quality comparison of edge areas interpolation.

The processing speed of the two-thread SIMD-optimized C implementation of the proposed method was 26 frames per second enlarging  $640 \times 352$  video 2 times at laptop based on Intel Core2 Duo P8400 2.2Ghz CPU.

## 5 Conclusion

In this paper we presented a novel local approach to the edge-directed image interpolation, based on a Bayesian framework. The flexibility of the technique allows the construction of different algorithm modifications, one of which was described. As the objective and subjective comparisons show, the algorithm outperforms conventional edge-oriented upscaling



Figure 8: Visual quality comparison of textured areas interpolation.

methods, combining high-quality edge interpolation with artifacts-free texture handling. The processing quality can be further increased by introducing "texture" and "uniform area" pixel types instead of the current "non-edge" type. This will make possible the usage of large interpolation kernels, *e.g.* lanczos, for textured areas, and small kernels (bilinear or bicubic) for uniform areas. Increasing the number of allowed pre-defined edge directions will also benefit.

The method suits well for the hardware implementation, particularly on massively parallel systems, *e.g.* modern GPUs.

## References

- [1] S. Farsiu, M.D. Robinson, M. Elad, and P. Milanfar. Fast and robust multiframe super resolution. *IEEE Trans. Image Processing*, 13(10):1327–1344, October 2004.
- [2] W.T. Freeman, Jones T.R., and Pasztor E.C. Example-based super-resolution. *IEEE Computer Graphics and Applications*, 22(2):56–65, March/April 2002.
- [3] J.W. Hwang and H.S. Lee. Adaptive image interpolation based on local gradient features. *IEEE Signal Processing Letters*, 11(3):359–362, March 2004.
- [4] M. Li and T.Q. Nguyen. Markov random field model-based edge-directed image interpolation. *IEEE Trans. Image Processing*, 17(7):1121–1128, July 2008.
- [5] X. Li and M.T. Orchard. New edge-directed interpolation. *IEEE Trans. Image Processing*, 10(10):1521–1527, October 2001.
- [6] D.D. Muresan. Fast edge directed polynomial interpolation. In *ICIP*, pages II: 990–993, 2005.
- [7] G. Ramponi. Warped distance for space-variant linear image interpolation. *IEEE Trans. Image Processing*, 8(5):629–639, May 1999.
- [8] R.R. Schultz and R.L. Stevenson. A bayesian approach to image expansion for improved definition. *IEEE Trans. Image Processing*, 3(3):233–242, May 1994.
- [9] K. Simonyan, S. Grishin, D. Vatolin, and D. Popov. Fast video super-resolution via classification. In *ICIP*, pages 349–352, 2008.
- [10] H. Takeda, S. Farsiu, and P. Milanfar. Kernel regression for image processing and reconstruction. *IEEE Trans. Image Processing*, 16(2):349–366, February 2007.
- [11] Q. Wang and R.K. Ward. A new orientation-adaptive interpolation method. *IEEE Trans. Image Processing*, 16(4):889–900, April 2007.