Comparison of Doubling the Size of Image Algorithms

S. E. Vaganov*, * and S. I. Khashin* , **
*Ivanovo State University, Ivanovo, 153025 Russia
**e-mail: pro100-pioner@mail.ru

e-mail: khash2@mail.ru

Received April 18, 2016

Abstract—In this paper the comparative analysis for quality of some interpolation non-adaptive methods of doubling the image size is carried out. We used the value of a mean square error for estimation accuracy (quality) approximation. Artifacts (aliasing, Gibbs effect (ringing), blurring, etc.) introduced by interpolation methods were not considered. The description of the doubling interpolation upscale algorithms are presented, such as: the nearest neighbor method, linear and cubic interpolation, Lanczos convolution interpolation (with \( a = 1, 2, 3 \)), and 17-point interpolation method. For each method of upscaling to twice optimal coefficients of kernel convolutions for different down-scale to twice algorithms were found. Various methods for reducing the image size by half were considered the mean value over 4 nearest points and the weighted value of 16 nearest points with optimal coefficients. The optimal weights were calculated for each method of doubling described in this paper. The optimal weights were chosen in such a way as to minimize the value of mean square error between the accurate value and the found approximation. A simple method performing correction for approximation of any algorithm of doubling size is offered. The proposed correction method shows good results for simple interpolation algorithms. However, these improvements are insignificant for complex algorithms (17-point interpolation, Lanczos \( a = 3 \)). According to the results of numerical experiments, the most accurate among the reviewed algorithms is the 17-point interpolation method, slightly worse is Lanczos convolution interpolation with the parameter \( a = 3 \) (see Table 2).

Keywords: interpolation, convolution of function, Lanczos filter, 17-point interpolation

INTRODUCTION

Image scaling is one of the classic tasks of computer graphics [2, 5, 10]. A special case of this problem is to reduce and double the image size. These methods play a special role in the compressing raster images [1, 7, 8, 10], where the quality of approximation depends on the overall efficiency of encoding.

The paper presents a comparative analysis of the error of some interpolation methods of doubling the image size. The standard deviation was used as an estimate of the accuracy of the algorithm. Only non-adaptive methods were taken into account, introduced artifacts (aliasing, blurring, etc.) were not considered. A simple method of “correction” (p. 1, 5) is proposed, which gives an approximation improvement on simple algorithms.

Let \( a(x, y) \) be the brightness matrix of a gray image or one of the color components of a full-color image, \( m_x \times m_y \) its dimensions. For convenience, we assume that the values of \( a(x, y) \) are defined for all pairs of integers \((x, y)\) by the formula

\[
a(x, y) = a(x', y'),
\]

where

\[
x' = \begin{cases} 
0 & \text{if } 0 < x, \\
x & \text{if } 0 \leq x < mx, \\
mx - 1 & \text{if } x \geq mx,
\end{cases}
\]

\[
y' = \begin{cases} 
0 & \text{if } 0 < y, \\
y & \text{if } 0 \leq y < my, \\
my - 1 & \text{if } y \geq my.
\end{cases}
\]
In addition, we can consider the matrix \( a[x, y] \) as a function of \( a(x, y) \) of two integer variables and extend its values to the entire real plane using some interpolation formula. Thus, the function \( a : \mathbb{R}^2 \rightarrow \mathbb{R} \) is defined on the entire real plane.

To obtain an image with a matrix \( b[x, y] \) half the size of \( \frac{m_x}{2} \times \frac{m_y}{2} \), you can use one of two basic methods.

(1) **Averaging:**

\[
b[x, y] = \frac{a[2x, 2y] + a[2x + 1, 2y] + a[2x, 2y + 1] + a[2x + 1, 2y + 1]}{4}.
\]

(2) **Even points:**

\[
b[x, y] = a[2x, 2y].
\]

In the present work we shall confine ourselves to the first method.

In some tasks (e.g. image compression), the size reduction algorithm may use a larger number of adjacent points. There is a need instead of (1) to consider a more general formula:

\[
b[x, y] = \sum_{i,j} a[2x + i, 2y + j] \cdot \beta(i, j),
\]

where \( \beta(i, j) \) a some weight function.

According to the measurement results, it can be noted that the transition from averaging over four points to a weighted average of 16 shows a noticeable improvement in the accuracy of approximation. Weighting at 36 gives some more improvement, but a further increase in the area has almost no effect on the results of interpolation. Therefore, in the present work, we limited ourselves to the case of 36 nonzero coefficients of the function \( \beta(i, j) \), among which only 6 are different. They can be arranged as follows:

\[
\begin{array}{cccccc}
\beta_5 & \beta_4 & \beta_3 & \beta_3 & \beta_4 & \beta_5 \\
\beta_4 & \beta_2 & \beta_1 & \beta_2 & \beta_4 \\
\beta_3 & \beta_0 & \beta_0 & \beta_1 & \beta_3 \\
\beta_1 & \beta_0 & \beta_0 & \beta_0 & \beta_1 \\
\beta_4 & \beta_2 & \beta_1 & \beta_2 & \beta_4 \\
\beta_3 & \beta_4 & \beta_3 & \beta_4 & \beta_3 \\
\end{array}
\]

and their sum should be equal to 1:

\[
4\beta_0 + 8\beta_1 + 4\beta_2 + 8\beta_3 + 8\beta_4 + 4\beta_5 = 1.
\]

The work consists of four parts. The first section describes some algorithms of linear image size doubling (nearest neighbor method, linear and cubic interpolation, interpolation by convolution with Lanczos kernel), their properties, as well as a description of the correction algorithm and 17-point interpolation method. The second section includes a description of the methods of optimal image size reduction for the doubling methods discussed in the first section. The third section contains the results of numerical estimates of the quality (accuracy) of the methods of doubling considered in the work. The last section summarizes the results of the work.

### 1. METHODS OF LINEAR-DOUBLING THE IMAGE

If the matrix \( b[x, y] \) of half size was obtained by the formula (1) from the matrix \( a[x, y] \), then

\[
a(x, y) = b(0.5x - 0.25, 0.5y - 0.25),
\]

and if by the formula (2),

\[
a(x, y) = b(0.5x, 0.5y).
\]

To approximate the value of the function at a point with non-integer coordinates, we will use convolution with some kernel \( R(x, y) \):

\[
f(x, y) = \sum_{i,j \in \mathbb{Z}} f(i, j)R(x - i, y - j).
\]
It is usually assumed that the kernel satisfies the conditions:

— \( R(x, y) = 0 \), if \( |x| \geq Q, |y| \geq Q \) for some constant \( Q \);
— \( R(x, y) = R(|x|, |y|) \);
— \( R(x, y) = R(y, x) \);
— \( R(0, 0) = 1 \);

To these conditions often add another, which is formalized as

\[
\sum_{i,j \in \mathbb{Z}} R(x - i, y - j) = 1,
\]

for all real \( (x, y) \). Obviously, it suffices to verify this equality for \( 0 \leq x, y \leq 1 \).

We will call the kernel \( R(x, y) \) **decomposable** if

\[
R(x, y) = R(x)R(y).
\]

When applying the formula (ref approx1), we need the values of the kernel function only at points of the form

\[
(i + 0.25, j + 0.25), (i + 0.75, j + 0.25), (i + 0.25, j + 0.75), (i + 0.75, j + 0.75)
\]

for integers \((i, j)\). In all the examples considered below, the constant \( Q \) does not exceed 3, therefore, in the case of a decomposable kernel, it is sufficient to know the values of the function \( R(x) \) in points

\[
0.25, 0.75, 1.25, 1.75, 2.25, 2.75,
\]

that is, only six values.

**1.1. Trivial, Linear and Cubic Interpolations**

The following methods of interpolating brightness values are most popular (see, for example, [2], p. 406):

—Trivial interpolation (nearest neighbor method), that is, by the nearest integer point. The interpolation kernel:

\[
R_0(x) = \begin{cases}
1 & 0 \leq |x| < 0.5, \\
0 & 0.5 \leq |x|.
\end{cases}
\]

—Linear interpolation, kernel

\[
R_1(x) = \begin{cases}
1 - |x| & 0 \leq |x| < 1, \\
0 & 1 \leq |x|.
\end{cases}
\]

—Cubic interpolation, its (one-dimensional) kernel can be written as

\[
R_3(x) = \begin{cases}
(1 - |x|)(2 - |x|)(3 - |x|)/6 & 0 \leq |x| \leq 1, \\
0 & |x| > 2.
\end{cases}
\]

**1.2. Lanczos Kernel**

The Lanczos kernel is considered one of the most effective in the interpolation of the brightness values of the images (see, for example, [6], p. 231–232).

Let

\[
sin \pi x
\]

\[
\begin{cases}
\sin(\pi x) & x \neq 0, \\
\pi x & x = 0.
\end{cases}
\]
The Lanczos function with the natural parameter $a$ (window size) is called the function

$$L(x)\begin{cases} C \sin c(x) \sin c\left(\frac{x}{a}\right) & |x| < a, \\ 0 & |x| \geq a, \end{cases}$$

where the constant $C$ is chosen such that

$$\int L(x) = 1.$$  

$a$   
$C$   
1 1.1076364143517988223
2 0.9903050711891360020
3 1.0029533506415752856

When using the Lanczos kernel, it should be borne in mind that due to the zeroing of its values at $|x| > a$, the condition $\sum_i R(x - i) = 1$ will be fulfilled only approximately. Therefore, the kernel weights must be normalized so that their sum is 1.

### 1.3. Decomposable Kernels

As mentioned above, in the case of a decomposable kernel, to find a matrix of doubled size, we do not need the entire function $R(x)$, but only six of its values:

$$R(0.25),\ R(0.75),\ R(1.25),\ R(1.75),\ R(2.25),\ R(2.75),$$

that is, in fact, a vector of six numbers. We explicitly write this vector in all cases of interest to us.

—Trivial interpolation, that is, by the nearest integer point. The values of the decomposable function $R_0$ in points (6) are as follows:

$$(1, 0, 0, 0, 0, 0).$$

—Linear interpolation, function values at six points:

$$(3/4, 1/4, 0, 0, 0, 0).$$

—Cubic interpolation:

$$(105, 35, -7, -5, 0, 0)/128.$$  

—Interpolation 5th degree:

$$(3465, 1155, -693, -495, 77, 63)/8192.$$  

—Lanczos kernel with parameter $a = 1$

$$(0.9, 0.1, 0, 0, 0, 0).$$

—Lanczos kernel with parameter $a = 2$

$$(0.8686065442, 0.2330001887, -0.08388006799, -0.01772666415, 0, 0),$$
or, rounding to $1/256$:

$$(222, 60, -21, -5, 0, 0)/256.$$  

—Lanczos kernel with parameter $a = 3$

$$(0.89277077, 0.27101057,-0.13327464,-0.067997263,0.030112286,0.0073782709),$$
or, rounding to $1/256$:

$$(229, 69, -34, -18, 8, 2)/256.$$  

**Remark 1.** The coefficients in the Lanczos kernels normalized so that they sum to one.

These coefficients can be selected so as to minimize the error on real files. Such sets will be called optimal.

—Optimal linear interpolation:

$$(7/8, 1/8, 0, 0, 0, 0).$$
—Optimal cubic interpolation:

\[(254, 48, -38, -8, 0, 0)/256.\]

—Optimal interpolation of 5th degree:

\[(256, 37, -36, -6, 4, 1)/256.\]

### 1.4. 17-Point Interpolation

In all the cases considered above, the kernel \( R(x, y) \) had the form of a product: \( R(x, y) = R(x)R(y) \). In general, this is not necessarily the case. In [4], a 17-point interpolation formula is proposed whose kernel cannot be written in this form.

If we want to use the kernel function only to double the size of the image, it will be enough to define the values of the function \( R \) at points of the form \( R(k \pm 0.25, l \pm 0.25) \) where \( k, l \) are integers. Therefore, taking into account the symmetry \( (R(x, y) = R(|x|, |y|)) \) we need the values of the function \( R(x, y) \) can be set by Table 1.

Thus, the function \( R(x, y) \) is determined by ten coefficients. It is named 17-point formula because the interpolated value is calculated as a weighted sum of the values of the original function in the 17 nearest integer points.

A large number of numerical experiments made it possible to find a formula close to optimal in this class:

\[\{r_0, \ldots, r_9\} = \{256, 45, 13, -46, -7, -9, 3, -2, 1, 10\}/256. \tag{7}\]

Adding additional nonzero coefficients to the kernel improves the quality of the algorithm by hundredths of a percent at best and can hardly be considered appropriate.

### 1.5. Doubling Algorithm Correction

Let \( B \) be the image halved with (1). Suppose we have some algorithm for doubling the image size \( \psi: B \rightarrow C \). If the enlarged image \( C \) is halved, then we do not necessarily get \( B \) when it is enlarged.

Therefore, consider the following algorithm to improve the restored image \( C \). Let \( B' \) be an approximation of the image \( B \) obtained by averaging the enlarged image \( C \). Then, due to linearity (1), the correction of \( C \) values is determined by the following relation:

\[
\begin{align*}
C(2x, 2y) &= C(2x, 2y) + e(x, y), \\
C(2x + 1, 2y) &= C(2x + 1, 2y) + e(x, y), \\
C(2x, 2y + 1) &= C(2x, 2y + 1) + e(x, y), \\
C(2x + 1, 2y + 1) &= C(2x + 1, 2y + 1) + e(x, y),
\end{align*}
\tag{8}
\]

where \( e(x, y) = B(x, y) - B'(x, y) \) is the value of the deviation of the true value from the approximation.

The presented correction method allows improving the quality of approximation of some linear doubling methods (see Table 2).
2. THE OPTIMAL IMAGE DOWNSCALING

In the case of a more difficult image downscaling interpolation formula (3) compared to (1), we can find the following six optimal coefficients

\[(\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5)\]  

so that the interpolation error is minimal for all above described methods. Consider these methods in more detail.

**Linear interpolation.** The found optimal coefficients \(\beta_i\):

\[(109, 0, -3, -30, 5, 8)/256.\]

**Cubic interpolation:**

\[(82, 8, -6, -20, 5, 2)/256.\]

**5th degree polynomial interpolation:**

\[(74, 8, -1, -17, 3, 3)/256.\]

**Interpolation with Lanczos kernel \(a = 1\):**

\[(77, 6, -2, -16, 4, 1)/256.\]

**Interpolation with Lanczos kernel \(a = 2\):**

\[(75, 7, -2, -16, 4, 1)/256.\]

**Interpolation with Lanczos kernel \(a = 3\):**

\[(64, 12, -3, -14, 3, 1)/256.\]

**Optimal linear formula.**

We was found two coefficients for a linear interpolation formula \((R(1/4), R(3/4))\) and six \(\beta_i\) factors. The found optimal kernels:

\[(R(1/4), R(3/4)) = (200, 56)/256\]

and

\[(\beta_0, \ldots, \beta_2) = (103, 0, -1, -26, 2, 10)/256.\]

**Optimal cubic formula.** The found four coefficients for a linear interpolation formula \((R(1/4), R(3/4), R(5/4), R(7/4))\) and \(\beta_i\) factors:

\[(R(\ldots)) = (235, 47, -23, -3)/256,\]

and

\[(\beta_0, \ldots, \beta_3) = (70, 7, 0, -12, 0, 4)/256.\]

| Method                  | \(D\)   | \(corr\) | \(ods\) | \(ods + corr\) |
|------------------------|---------|----------|---------|----------------|
| Trivial                | 10.10817|          |         |                |
| Linear                 | 0.91519 | 0.82115  | 0.79500 | 0.78666        |
| Cubic                  | 0.84574 | 0.79950  | 0.78953 | 0.77392        |
| 5th degree             | 0.82708 | 0.79283  | 0.78304 | 0.76908        |
| Lanczos \(a = 1\)      | 0.92254 | 0.90744  | 0.90265 | 0.90601        |
| Lanczos \(a = 2\)      | 0.82749 | 0.80067  | 0.78471 | 0.77886        |
| Lanczos \(a = 3\)      | 0.80676 | 0.79032  | 0.77608 | 0.76897        |
| Optimal linear         | 0.91224 | 0.88835  | 0.80943 | 0.78512        |
| Optimal cubic          | 0.79730 | 0.79565  | 0.79589 | 0.77739        |
| Optimal 5th degree     | 0.80825 | 0.80825  | 0.78501 | 0.77580        |
| 17 points formula      | 0.79032 | 0.79032  | 0.76898 | 0.76890        |
Optimal 5th degree formula. The found optimal six coefficients for a linear interpolation formula
\[(R(1/4), R(3/4), R(5/4), R(7/4), R(9/4), R(11/4))\]
and \(\beta_i\):
\[(R(\ldots)) = (236, 54, -49, -7, 21, 1)/256\]
and
\[(\beta_0, \ldots, \beta_5) = (62, 14, 3, -13, -2, 1)/256.\]

17-points formula. The found optimal coefficients:
\[\{\beta_0, \ldots, \beta_5\} = (72, 15, 2, -17, -4, 2)/256,\]
and interpolation formula kernel \(r_0, \ldots, r_5\):
\[\{190, 45, 14, -34, -4, -2, 6, 1, 16\}/256.\]

3. CALCULATION RESULTS

When comparing the algorithms, a large number of images of different origin and quality were tested. The table below shows the results for 13 small standard images often used in this kind of experiments:

- Airplane.bmp, Car.bmp, Couple.bmp, Girl.bmp, House.bmp, Lake.bmp, Lena.bmp, Mandrill.bmp, Peppers.bmp, Splash.bmp, Tiffany.bmp, Tree.bmp, Woman.bmp.

Let \(B\) be a halved image and \(C_T\) be an image \(B\) enlarged by the trivial method. Then the quality assessment of the doubling algorithm can be represented by the ratio:

\[D(A, C) = \frac{MSE(A, C)}{MSE(A, C_T)},\]  \hspace{1cm} (10)

where \(A\) is the original image, and \(C\) is the image \(B\) doubled by the method under consideration.
For each method, the table shows the averaged values (10) across the entire set of files in four variants:

1. The original method.
2. With “corr” correction.
3. With the optimal reduction of “ods”.
4. With optimal reduction and correction of “ods + corr”.

For example, for the “Lanczos $a = 3$” method, the error is 88.289% of the error of the trivial method, and under the condition of optimal reduction and subsequent correction – 83.809% of the error of the trivial method.

The evaluation in other images, of course differ from those in the table, but not too much. And most importantly, the ratio between the different methods remains the same.

4. CONCLUSIONS

—If the image reduction is carried out by averaging over 4 adjacent points, the best results among the considered linear formulas are obtained by 17-point interpolation with the kernel (7). Further complication of the formula improves the result by hundredths of a percent and can hardly be considered expedient.

—Correction of the doubled image (par. 1.5) — the proposed method of increasing the accuracy of approximation. It gives excellent results for simple methods (e.g. linear interpolation). However, when using the most complex algorithms (17-point, Lanczos $a = 3$) the improvements are negligible.

—If it is possible to choose the method of halving the image or the algorithm of reducing by which the original image $A$ was obtained is known, then the doubling algorithm can be somewhat improved.

—Further improvement of accuracy is possible only with the use of nonlinear or adaptive methods.

FUNDING

This work was supported by the Federal targeted Program “Scientific and Scientific-pedagogical Personnel of Innovative Russia” under the grant no. 162 (2014/40).

CONFLICT OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

1. Vatolin, D., et al., Metody szhatiya dannykh. Ustroistvo arkhivatorov, szhatie izobrazhenii i video (Data Compression Methods. The Structure of Archivers, Compression of Images and Videos), Moscow: Dialog–MIFI, 2002.
2. Gonzalez, R.C. and Woods, R.E., Digital Image Processing, Prentice Hall, 2002.
3. Pratt, W.K., Digital Image Processing, Wiley, 1978.
4. Khashin, S.I., 17-point interpolation formula of two variables, Vestn. Ivanov. Gos. Univ., 2003, no. 3, pp. 133–137.
5. Jahne, B., Digital Image Processing, Springer, 2005.
6. Burger, W. and Burge, M.J., Principles of Digital Image Processing: Core Algorithms, London: Springer-Verlag, 2009.
7. Taubman, D.S. and Marcellin, M.W., JPEG2000: Image Compression Fundamentals, Standards, and Practice, Springer Science+Business Media, LLC, 2002.
8. Wallace, G.K., The JPEG still picture compression standard, IEEE Trans. Consum. Electron., 1992, vol. 38, no. 1, pp. xviii–xxxiv.
9. Keys, R.G., Cubic convolution interpolation for digital image, Proc. IEEE Trans. Acoust. Speech Signal Process., 1981, vol. 29, no. 6, pp. 1153–1160.
10. Klette, R., Concise Computer Vision. An Introduction into Theory and Algorithms, London: Springer-Verlag, 2014.
11. Xin, Li and Orchard, M.T., New edge-directed interpolation, IEEE Trans. Image Process., 2001, vol. 10, no. 10, pp. 1521–1527.