Triangle based Image Magnification

V. Vyšniauskas

Faculty of Technology, Siauliai University,
Vilniaus str. 141, LT-76353 Siauliai, Lithuania, phone: +370 41 595835; e-mail: vytautas.vysniauskas@tf.su.lt

Introduction

Image magnification is a process of obtaining an image at resolution higher than taken from image sensor. Image magnification synonyms are interpolation, enlargement, zooming, etc. To create higher resolution image, previous image must be complemented with new pixels and their intensity must be calculated.

Commonly, image magnification is accomplished through convolution of the image samples with a single kernel – typically the nearest neighbour [1, 2] bilinear, bicubic [3] or cubic B-spline kernel [4] interpolation and training-based algorithms [5, 6].

Triangle based image magnification method

Intensity surface can be explained as landscape with hills and hollows, rides and valleys. Any 3 dimensional (3D) surfaces, with some approximation, can be created from triangles. Look at the image intensity as 3D surface build from triangles where vertexes are image pixels with intensity as z axis (Fig. 1).
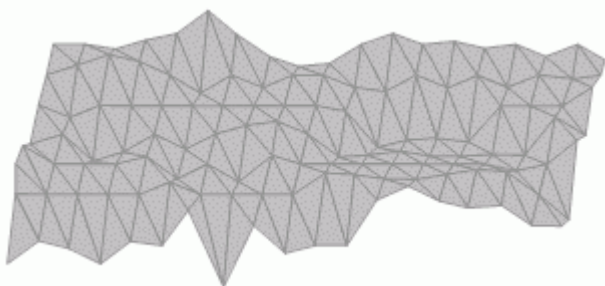

Fig. 1. Image intensity surface

To magnify image, new image must be complemented with new pixels, added necessary columns and rows of pixels, and calculated new pixels intensity. In this work is made assumption, that new pixels intensity is situated on suitable triangle planes (Figure 1).

During image magnification contours are blurred because of slope sharpness reduction. This is the main problem for all known magnification methods.

Look at small part of intensity surfaces in Figure 2. It is evident, that, in this situation, sharpness of slope is better when triangles are created like in Figure 2b than in Figure 2a. Accordingly the base of both triangles must be diagonal with less intensity gradient. That partially reduce magnification blur.
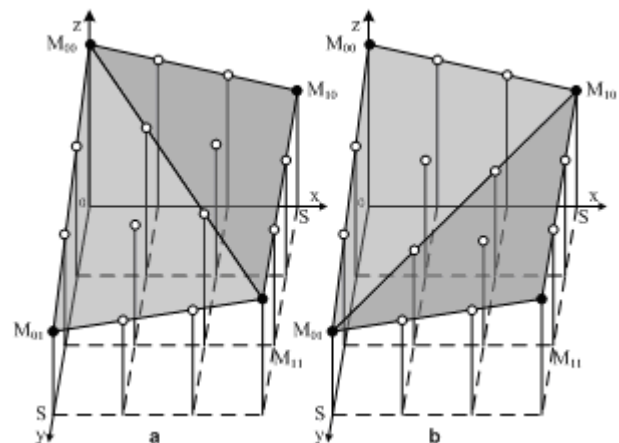

Fig. 2. Two variants of creating pairs of triangles via four pixels

So must be located distinctive points on intensity surface and different magnification algorithm applied. There are two things that must be solved: choose the best arrangement of triangles and calculate new pixels intensity.

To explain triangle based magnification method, get four neighbouring pixels from original image and create two triangles planes throws these pixels as vertexes (black dots in Figure 2). These pixels 3D coordinates is known, because they are taken from original image. White dots are new pixels that were inserted to magnify image. Their intensity must be calculated.

Early was decided, that new pixels will situated on two triangles planes. Triangle plane can be formulated as equation of three coplanar vectors over spatial points $M_{00}(0,0,Z_{00})$, $M_{10}(S,0,Z_{10})$, $M_{01}(0,S,Z_{01})$ where x and y coordinates are related with magnification S.

$$\begin{bmatrix} x & y & z - Z_{00} \\ S & 0 & Z_{10} - Z_{00} \\ 0 & S & Z_{01} - Z_{00} \end{bmatrix} = 0 , \qquad (1)$$

where x, y, z - spatial pixel coordinates, S - magnification, Z_{xy} - image pixel with coordinates (x, y) intensity.

Pixel intensity that is under triangle plane derived from matrix (1) as equation:

$$z_{xy} = \left((Z_{10} - Z_{00}) \cdot x + (Z_{01} - Z_{00}) \cdot y + S \cdot Z_{00} \right) / S . \qquad (2)$$

Intensity equation (2) can be formalized as function of pixel coordinates

$$z_{xy} = F_{nm}(x, y), \qquad (3)$$

where nm is index of vertex, in front of base, to mark triangle plane.

As show in Fig. 2, there are two different ways to draw pair of triangles and calculate pixels intensity.

When magnification S is integer, x and y can get discrete integer values from 0 to S. To calculate pixels, with coordinates (x, y), intensity are used formulas (4, 5). When triangles are situated as shown in Fig. 2a,

$$z_{xy} = \begin{cases} F_{10}(x, y), & y \leq x; \\ F_{01}(x, y), & y > x. \end{cases} \qquad (4)$$

Or when triangles are situated as shown in Figure 2b

$$z_{xy} = \begin{cases} F_{00}(x, y), & (S - y) \leq x; \\ F_{11}(x, y), & (S - y) > x. \end{cases} \qquad (5)$$

To decide what is a better way to situate triangles planes, was used intensity vector direction that direction is from pixel with lowermost intensity to pixel with topmost intensity of four pixels (M_{00}, M_{10}, M_{01} and M_{11}). Vector can takes eight discrete directions and was marked with number that explained in Figure 3.
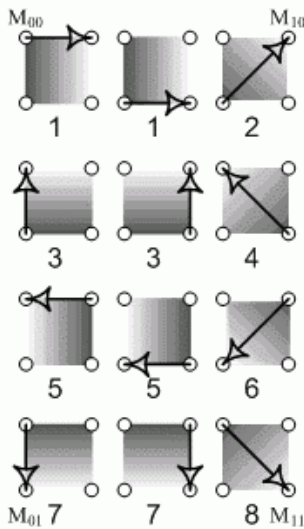

Fig. 3. Vector direction marks

Vector mark is main characteristic that allow solving what variant, of two shown in Fig. 2, is better. For example, when vector value is 2, the best way to situate triangles is shown in Fig. 2a, in other hand, when vector value is 4, the best way to situate triangles is shown in Fig. 2b. In comparison of triangles in both of Fig. 2 pictures, is evident that in Fig. 2b, sharpen of triangle plane (M_{00}, M_{10}, M_{01}) will increased and plane (M_{11}, M_{10}, M_{01}) became flatten, so magnified image sharpness will slightly increase.

Testing

This image magnification method was implemented as C++ function in conjunction with free image processing toolkit CImg. For compressed image processing was used Image Magic package.

Scaling down and magnify factors were selected integer numbers. For example scaling factor 3 means that both width and height of image will be divided or multiplied by 3.

Naturally magnified image has some distortions. To measure distortions was calculated root-mean-square-error (RMSE) between original and magnified images. First of all original image was scaled down with scale factor, then image was magnified with the same scale factor, and after that calculated RMSE between original and magnified images for each pixel on each colour layer R, G and B.

For comparison, magnification was executed with three well known magnification methods: block, linear, bi-cubic interpolation and introduced triangle based magnification that will be call "Triangle".

Two different types of compressed digital image formats JPG and PNG were used. JPG is lossy, but very popular, image format and PNG is lossless format.

For there first test series were selected 9 JPG and 9 PNG images with different size and significant difference in number of small and large objects. Test series with scaling factor 2, 3, 4, 5, 6, 7, 8 was produced with each image.

The second test was the same but with 5000 images with various size and format types for statistic analysis.

Test results

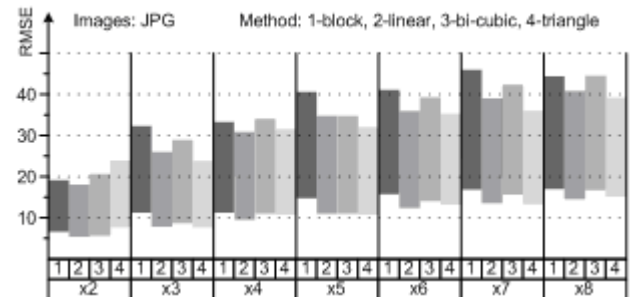First test series of image magnification RMSE is shown in Fig. 4 for JPG images and Fig. 5 for PNG images.

Fig. 4 JPG image magnification RMSE

Because of significant difference of RMSE for different images in figures are shown difference from
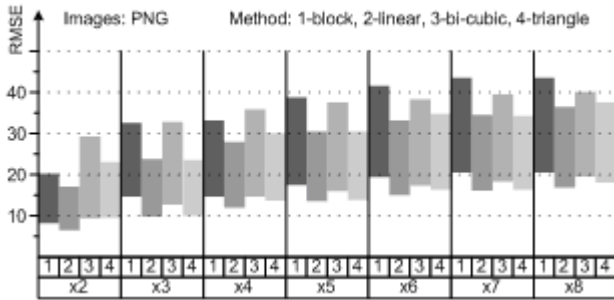

Fig. 5 PNG image magnification RMSE

minimal to maximal RMSE values of tested images for different methods and magnification factors. As seen from figures (Fig. 4. and Fig. 5.) "Triangle" magnification method quality is nearly linear interpolation quality (less RMSE is better).
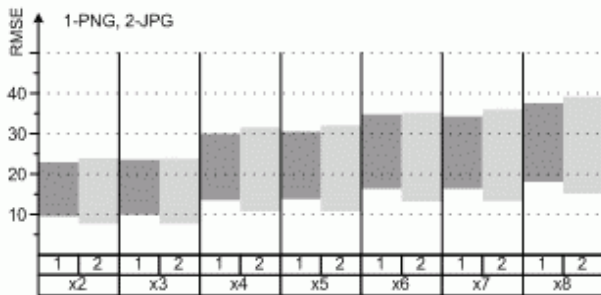

Fig. 6 PNG and JPG image magnification RMSE

"Triangle" magnification method RMSE of JPG and PNG image formats is shown in Fig 6. PNG image magnification distortion is slightly less, because JPG compression add some distortion in to image.

Fig. 7. Motocross

Fig. 8. Belle

Some thing must be explained about RMSE diagrams. For example picture (Fig. 7) RMSE was biggest for all magnification methods and all scaling factors, and picture (Fig. 8) RMSE was least for the same conditions. So great difference in RMSE (from 15 to 25) was found because for testing were selected images with different amount of small objects. For example break stones or motocross (Fig. 7) images that have biggest RMSE, have a great amount of

small details. Landscape, space or human face (Fig. 8) images with big flat areas have smallest RMSE.

There was calculated ANOVA Statistical Significance (p-level) between triangle and each other three methods, that where used for magnification methods comparison (Table 1), with second test series. In Table 1 is shown average RMSE values. Less RMSE is better and Triangle and Linear methods average RMSE are near the same.

Table 1. RMSE statistics of magnification method

	Triangle	**Linear**	**Block**	**Bi-cubic**
Count	31200	31200	31200	31200
Average	19.865	19.803	24.099	22.898
Standard deviation	10.146	10.035	11.939	11.492
P-value	-	0.443	0	1.0E-261

During magnification was measured magnification time for all tested methods and scaling factors. Magnification time was measured in milliseconds. Because of different size and coloured or black and white image, magnification speed was recalculated by formula:

$$P = \frac{width \cdot height \cdot colour}{t_m} \quad \left(\frac{pix}{ms} \right), \qquad (6)$$

where t_m – magnification time, colour – colour layer number (RGB=3, B/W=1). Number P in formula (6) is a magnification speed with dimension pixels per millisecond, and shows how many pixels of black and white (b/w) image are processing per one millisecond. For colour image that have had 3 colour layers processing time will increase 3 times.

Table 2. Average magnification speed pix/ms of b/w image

Magnification Method	Average time (pix/ms)			
	P4, 3 GHz		P3,750 MHz	
	JPG	PNG	JPG	PNG
Block interpolation	8013	7937	4618	4520
Linear interpolation	879	876	417	428
Bi-Cubic interpolation	969	971	426	435
Triangle	**6849**	**6897**	**3788**	**3436**

In Table 2 is shown average magnification speed (pixels per millisecond) of b/w image on PC Intel Pentium 4 3 GHz (P4 3 GHz) and Intel Pentium III 750 MHz (P3 750 MHz) for fist test series.
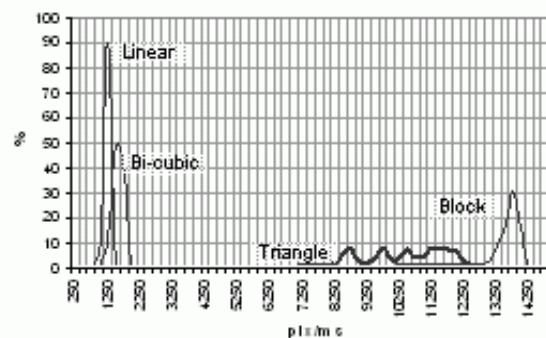

Fig. 9. Magnification speed histograms

Triangle method speed is more than 7 times faster than linear and more than 8 times than bi-cubic interpolation methods, with the nearly same quality and only 1.2 times slower than the simplest and fastest block magnification method. The same ratio of magnification time is on computers with different processor speed.

Magnification speed was calculated during second test series that time-span was a few days. Some dispersion was as result of computer multitasking and to get proper results, was build speed histograms that's show percents from samples count (Fig. 9).

In Fig. 9 are shown magnification speed histograms of 5000 pictures. Statistically triangle magnification speed is more than 8 times faster than linear or bi-cubic. Block, linear and bi-cubic interpolation have had strong peaks, but "Triangle" method have had four peaks.

That is because "triangle" method spends different time to make decision how to create triangles and spend different time for different magnification scale.

Conclusion

Triangle magnification method distortion is nearly linear interpolation method. When magnification factor grows, RMSE grows a bit slower than in linear or bi-cubic interpolation and significant slower than in block magnification (Fig. 4, 5).

Triangle magnification method is faster form 7 to 8 times with the quality nearly linear interpolation methods (Table 2 and Fig. 9).

Triangle magnification method can be implemented on microprocessor or microcontroller with only integer arithmetic, when magnification factor is integer.

References

1. **Gonzalez R. C., Woods R. E.** Digital Image Processing. – 2nd. Ed. – 2002, ISBN 0-201-18075-8. – P. 64–74.
2. **Seul M., O'Gorman L., Sammon M. J.** Practical Algorithms for Image Analysis. – 2001. ISBN 0-521-66065-3. – P. 43–67, 101–104.
3. **Netravali A. N., Haskell B. G.** Digital Pictures: Representation, Compression and Standards. – 2nd Ed. – New York: Plenum Press, 1995.
4. **Unser M., Aldroubi A., Eden M.** Fast B-spline Transforms for Continuous Image Representation and Interpolation // IEEE Trans. Pattern Anal. 1991, Mach. Int. – Vol. 13, No. 3. – P. 277–285.
5. **Freeman W. T., Jones T. R., Paszor E. C.** Example-Based Supper-Resolution // MERL, Mitsubishi Electric Research Labs, 201 Broadway Cambridge, MA 02139, August 2001.
6. **Candocia F. M., Principe J. C.** Superresolution of Images Base on Local Correlations // IEEE Transactions on Neural Networks. – March 1999. – Vol. 10, No. 2. – P. 372–380.

V. Vyšniauskas. Triangle based Image Magnification // Electronics and Electrical Engineering. – Kaunas: Technologija, 2006. – No. 6(70). – P. 45–48.
There was presented image magnification method based on new pixels intensity calculation from suitable triangles planes created via original image pixels as vertexes. This method was implemented as C++ function. Root-mean-square-error (RMSE) between original and magnified image was calculated for quality testing. To compare images original image size was reduced the same times as later magnified. Quality of presented method is nearly the same as of well known linear interpolation. Processing speed is from 7 to 8 times faster than in case of linear interpolation. When magnification coefficient is integer, this method can be implemented with integer mathematic on processors without floating point instructions. Ill. 9, bibl. 6 (in English; summaries in English, Russian and Lithuanian).

В. Вишняускас. Увеличение изображения на основе треугольников // Электроника и электротехника. – Каунас: Технология, 2006. – № 6(70). – С. 45–48.
Представлен метод увеличения изображения основан на вычислении яркости новых точек изображения из соображения, что они лежат на треугольных плоскостях, беря точки увеличиваемого изображения как вершины для построения треугольных плоскостей. Этот метод был реализован как функция на языке C++. Для оценки качества исчислялась средняя квадратическая ошибка всех точек начального изображения с увеличенным. Для сравнения изображений начальное изображение уменьшалось такое же число раз как потом увеличивалось. Результаты показали что качество схоже с известным методом линейной интерполяции. Скорость обработки от 7 до 8 раз выше по сравнению с линейной интерполяцией. Если увеличение производится в целое число раз, метод можно осуществить с помощью целочисленной математики и выполнять на процессорах не имеющих арифметики с плавающей запятой. Ил. 9, библ. 6 (на английском языке; рефераты на литовском, английском и русском яз).

V. Vyšniauskas. Vaizdo didinimas trikampių pagrindu // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2006. – Nr. 6(70). – P. 45–48.
Pristatomas vaizdo didinimo metodas, paremtas įterptų vaizdo taškų ryškio apskaičiavimu iš trikampių plokštumų taškų, kurios sudaromos naudojant didinamo vaizdo taškus kaip viršūnes. Šis metodas buvo realizuotas kaip C++ funkcija. Testavimui buvo skaičiuojama vidutinė kvadratinė paklaida tarp originalaus vaizdo ir padidinto. Vaizdams palyginti pradinis vaizdas prieš tai būdavo sumažinamas tiek kartų, kiek po to didinamas. Pristatomo metodo kokybė beveik tokia pat kaip žinomo tiesinės interpoliacijos metodo. Greitis nuo 7 iki 8 kartų didesnis nei anksčiau minėto tiesinės interpoliacijos metodo. Jeigu didinimo koeficientas yra sveikas skaičius, galima naudoti sveikųjų skaičių aritmetiką ir realizuoti procesoriuose ar kontroleriuose, kurie neturi slankiojo kablelio aritmetikos. Il. 9, bibl. 6 (anglų kalba; santraukos anglų, rusų ir lietuvių k.).