2008 International Conference on Computer Science and Software Engineering

A Novel Nonlinear Scaling Method for Video Images

Shi Zaifeng，Xiao Xia，Shao Kai and Jiao Yang
School of Electronic Information and Engineering, Tianjin University
No. 92 Weijin Road , Nankai District, Tianjin，China

***Abstract*—Digital Video image scaling is a full-fledged technology, and can be found everywhere in our daily life. However, the appearance of HDTV brought some problem between the new display devices and the old video signals. In this paper, a novel method of nonlinear scaling is proposed, aim to overcome the compatibility problem. Different from traditional linear approaches, in which the interpolate points are spaced evenly, the nonlinear approach split the whole image into several areas, each area is treated separately. So after scaling, the central area of the images can still keep the proportion, in the rest parts, calculated distortion can be found, according to human visual theory, it is acceptable. And the simulation result shows that the proposed approach is more efficient than conventional approaches.**

Keywords-image scaling; nonlinear interpolation; high definition;

I. INTRODUCTION

In recent years, HDTV(High Definition Television)[1] is becoming more popular everyday, following from that, high resolution LCDs and many other flat panel displays will also be widely spreaded before too long[2]. However, SDTV(Standard Definition Television)signals, in a considerable time, will be widely used[3]. So, there is a compatible problem when using displays whose aspect ratio is 16:9 however the video signals are 4:3(or 5:4)[4]. Usually, 3 approaches are used to solve the problem: (1) stretching the images to full screen size; (2) cutting the images to 16:9, then stretching them to suit the screen size; (3) adding blanking zones to the both sides of the images, then stretching them to suit the screen width [5]. But, all the approaches above have some problems, which, more or less, made them unacceptable. When stretched to full screen size, the original images are enlarged and stretched, and output images are deformed; Being cut to 16:9, the top part and the bottom part of the images are lost; adding blanking zone to the sides of the images make it impossible to watch the video in full screen mode, though it keeps the proportion of the image and all the pixel data of the original images.

In this paper, a method based on nonlinear interpolation is introduced[6]. First, the proposed method divide the original video images into 3 parts in the horizontal direction, marked L, M, R, respectively, and then decide the width of each part, compute the zoom factors of all the interpolate points. In the vertical direction, linear interpolation is used to scaling the images, since the scaling-up rate is less than that in the horizontal direction. For the area marked M, keep the width-high rate during scaling up, so it's called linear part; in part L and part R, nonlinear interpolation algorithm is used. The basic interpolation algorithm adopted in this approach is bi-cubic interpolation algorithm[7], in both directions, in order to get a good result.

II. NONLINEAR INTERPOLATION

In order to show 4:3(or 5:4) images on 16:9 displays, it is impossible to keep the proportion of the whole image and all the image data by using traditional linear interpolation algorithms. Different from traditional linear interpolation algorithms, nonlinear interpolation algorithm divides the images into 3 parts, and has different interpolate points density at different part of the images.

According to the space-based human visual attention theory[8], human attention is allocated to a location of the image, and at most of the time, the location is about in the middle of the image. So we divided the input video images into 3 parts as Fig.1 shows, and then scaling them by different methods. When watching TV or seeing a film, people usually pay most attention to the middle of the pictures, and pay less attention to the both sides of the pictures, so the distortion of the images in the nonlinear areas in Fig.1 can be acceptable.

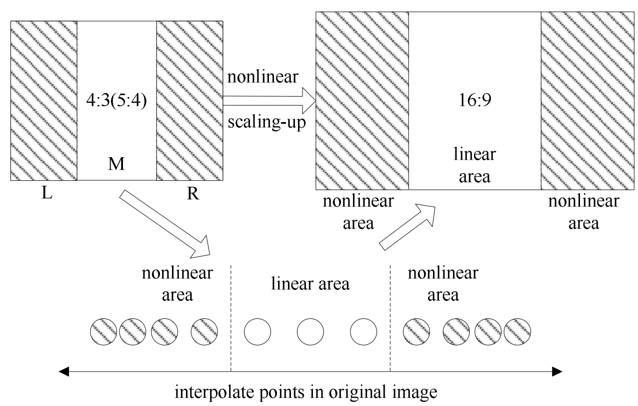


Fig.1 Nonlinear scaling-up(4:3(5:4) to 16:9)

We assume that every pixel in the new 16:9 image have a unique source point in the original image, we call the source points interpolate points. The interpolate points in M in original video images are evenly distributed; however, those in L and R are unevenly distributed. The horizontal distance of the interpolate points decreases as the position of the interpolate points get further from M. In the horizontal direction, the minimum distance is to get at the edges of the images, and maximum can be obtained on the border of L and M, and on the border of R and M, in order to make the transition near the border smooth. In horizontal direction, the interpolate points in input video images arrange as Fig.1 shows.

A. Divide the images

As is mentioned above, before scaling, the input video images should be divided into 3 areas, including 2 nonlinear

978-0-7695-3336-0/08 $25.00 © 2008 IEEE
DOI 10.1109/CSSE.2008.1227

areas and a linear area. So the first thing the algorithm must do is to decide the position of the border line, or the width of the linear area.

First, the zoom factor of the linear area should be found. As a linear area, the horizontal zoom factor equals to the vertical zoom factor. So, the zoom factor of the linear area is written by:

$$f_l = h_{new} / h \tag{1}$$

Where, f_l is the zoom factor of the linear area, h_{new} is the high of the new 16:9 images, and h is the high of the original 4:3(or 5:4) video images.

Second, the width of the linear area in the post scaling images can be figured out, which is:

$$W_{M(new)} = W \times R_M \times f_l \tag{2}$$

Where, $W_{M(new)}$ is the width of the post scaling area M, W is the width of the whole original images, and R_M is the ratio of the width of original linear area and the width of the whole original images.

Naturally, we can get the width of post scaling nonlinear areas by:

$$W_{L,R(new)} = (W_{new} - W_{M(new)}) / 2 \tag{3}$$

Where $W_{L,R(new)}$ is the width of the post scaling area L and R, and W_{new} is the width of the post scaling images. Area L and area R have the same width, and also, after scaling, the two nonlinear areas have the same width.

B. Find the rule the interpolate points are arranged in the horizontal direction

In this paper, the horizontal distance of the interpolate points is made to decrease by a fixed value, from the borders of linear area and nonlinear areas to the edges of the images, this means that the distances of the interpolate points in the horizontal direction constitute a equal difference progression, this can be see from Fig.2. So, if we can found the common difference and the first element of the progression, the coordinates of all the interpolate points can be fixed.

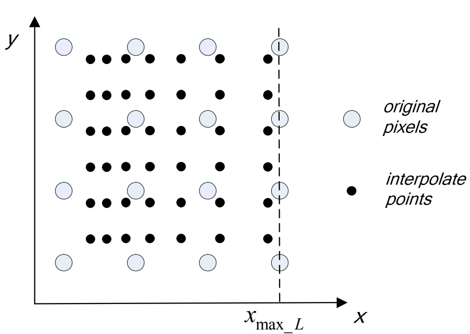


Fig.2 Interpolate points in L in original images

Assume that in the original images, the distance between adjacent pixels is 1. So the width of the image is $W-1$, and the distance between adjacent interpolate points in the linear area is $1/f_l$. The horizontal distance of the interpolate points at the borders of linear area and nonlinear areas is also $1/f_l$. So the width of nonlinear areas is:

$$W_{L,R} = [W - 1 - (W_M) \times (1/f_l)] / 2 \tag{4}$$

Where the W_M is the width of linear area in the original images.

The width of the nonlinear areas is written by:

$$W_{L,R} = \frac{\{(a_1 + d) + [a_1 + (W_{L,R(new)} - 1) \times d]\} \times (W_{L,R(new)} - 1)}{2} \tag{5}$$

Where the d is the common difference and a_1 is the first element of the equal difference progression.

According to the equal difference progression, at the border of nonlinear area and linear area, the distance of interpolate points can be written by:

$$a_1 + W_{L,R(new)} \times d = 1/f_l \tag{6}$$

From (4), (5) and (6), we can figure out the value of a_1 and d:

$$a_1 = \frac{[W - 1 - (W_{M(new)} / f_l)]}{W_{L,R(new)} - 1} - \frac{1}{f_l} \tag{7}$$

$$d = \frac{1/f_l - a_1}{W_{L,R(new)}} \tag{8}$$

C. Compute the coordinates of the interpolate points in the horizontal direction

So far, the range of the 3 areas are fixed, according to the value of a_1 and d, all the coordinates of interpolate points can be settled.

1)Part L:($W_{L,R(new)}$ rows, from 0 to $W_{L,R(new)} - 1$)

The horizontal coordinate of interpolate points in the n-th row x_{nl} in L is the sum of the n distances left, and can be calculated through the following formula:

$$x_{nl} = (a_1 + d) + (a_1 + 2d) + \cdots + (a_1 + nd) \tag{9}$$

A cut-off operate is necessary because sometimes the value of a_1 may be negative, the operation is simply set the value of a_1 to 0.

2)Part M:($W_{M(new)}$ rows, from $W_{L,R(new)}$ to ($W_{L,R(new)}$ + $W_{M(new)} - 1$))

The coordinate of interpolate points in the last row in L is:

$$x_{\max\_L} = (W_{L,R(new)} - 1)(a_1 - d/2) \tag{10}$$

And in linear area M, the interpolate points are evenly distributed, the horizontal distance of interpolate points is $1/f_l$, so the coordinate of the n-th row is:

$$x_{nm} = x_{\max\_L} + (n+1)/f_l \tag{11}$$

3)Part R:($W_{L,R(new)}$ rows, from ($W_{L,R(new)} + W_{M(new)}$) to (2$W_{L,R(new)} + W_{M(new)} - 1$))

The horizontal coordinate of the last raw in M is:

$$x_{\max\_M} = x_{\max\_L} + W_{M(new)}/f_l \tag{12}$$

The distribution of interpolate points in R are similar with those in L. The coordinate of the n-th row in area R is:

$$x_{nr} = x_{\max\_M} + 1/f_l + [a_1 + (W_{L,R(new)} - 1)d] + \cdots + [a_1 + (W_{L,R(new)} - n)d] \tag{13}$$

The coordinate of the last row should not greater than the value of W, so coordinate should be set to W if it does greater than W.

Interpolate points are evenly distributed long the vertical direction, so the vertical coordinate of the points in the n-th line is:

$$y = n/f_l \tag{14}$$

D. Compute the gray values of the interpolate points using bi-cubic interpolate algorithm

As a commonly used interpolation algorithm, bi-cubic interpolation algorithm works well in video image scaling.

The algorithm can be explained in Fig.3, the white balls are the original pixels in the original image, and the balls in deep color are interpolate points.

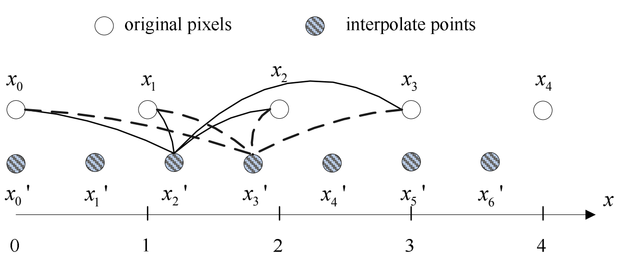


Fig.3 Bi-cubic interpolation algorithm(single dimension)

Take a for example, assume the coordinate of a is x, and y, $y_0 - y_3$ are gray values of a, $x_0 - x_3$, respectively, so:

$$y = \frac{(x-x1)(x-x2)(x-x3)}{(x0-x1)(x0-x2)(x0-x3)}y0 + \frac{(x-x0)(x-x2)(x-x3)}{(x1-x0)(x1-x2)(x1-x3)}y1 + \frac{(x-x0)(x-x1)(x-x3)}{(x2-x0)(x2-x1)(x2-x3)}y2 + \frac{(x-x0)(x-x1)(x-x2)}{(x3-x0)(x3-x1)(x3-x2)}y3 \tag{15}$$

III. SIMULATION AND ANALYSIS

The procedure of horizontal scaling is different from that of vertical scaling, so we split the whole process into two main parts: horizontal scaling and vertical scaling, their central module is scale_line and scale_hight, respectively. The whole hardware architecture is show in Fig.4.

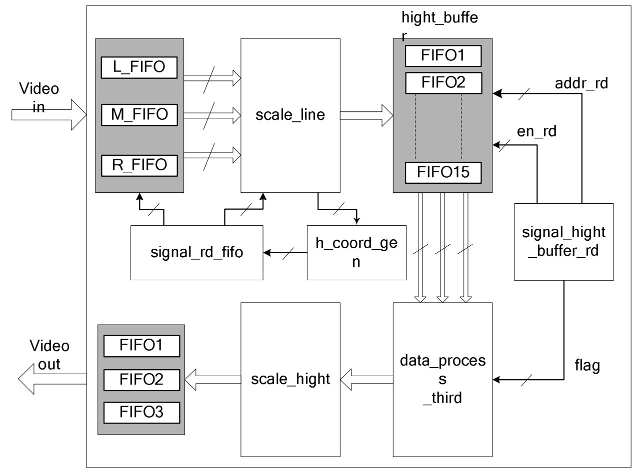


Fig.4 Hardware architecture of nonlinear scaling

Every single line of input video data are splited into 3 sections based on the widths of the 3 areas which are computed beforehand. The h_coord_gen module generates a horizontal coordinate of a new row, from 0 to $W_{new} - 1$, and output the coordinate to signal_rd_fifo module, after that, the signal_rd_fifo module decide whether it is necessary to read data from the 3 FIFOs, and if it is necessary, then from which FIFO, the data should be read.

For example, if the last coordinate x_n is between $W_{\max\_L}$ and $W_{\max\_L} + 1$, and the coordinate of the new row x_{n+1} is between $W_{\max\_L} + 1$ and $W_{\max\_L} + 2$, then the scale_line module should read in new data from M_FIFO. Since the gray value of the interpolate points in row x_n grows out of the values of pixels in rows $W_{\max\_L} - 1$, $W_{\max\_L}$, $W_{\max\_L} + 1$, and $W_{\max\_L} + 2$, and for the new row x_{n+1}, pixels in row $W_{\max\_L}$, $W_{\max\_L} + 1$, $W_{\max\_L} + 2$ and $W_{\max\_L} + 3$ is needed, so the data in row $W_{\max\_L} + 3$ should be read in, and the data in row $W_{\max\_L} + 3$ is stored in M_FIFO, so the scale_line module will read the data from M_FIFO.

After horizontal scaling, the output data is sent to hight_buffer, where 5 lines data can be saved at most, in order that the scale_hight module can work correctly. Here, the signal_ram_third_rd module and data_process_third module are used to adjust and control the input of scale_hight module, and make the data flow in succession.

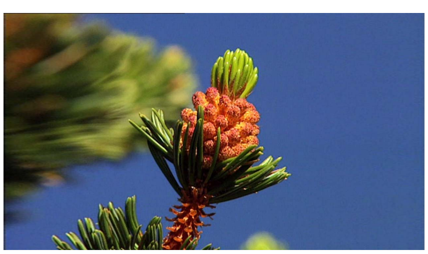
The output image is show in Fig.5, where the original image and the results of other approaches are given as well. Our approach is well able to keep the proportion of the main area of the image and all the pixel data in the original image, the distortion in the output image is accepted.

(a)original image

(b)expand mode

(c)crop mode

(d)letter-box mode



(e)Proposed method

Fig.5 Comparison of image scaling approaches

In order to show the result objectively, and make it easy to distinguish the difference between nonlinear scaling and expand mode, we scaled another image, which has only a circle in the middle of it. Fig.6 shows parts of the up-scaled images, now it's not very difficult to tell the advantage of the proposed method.

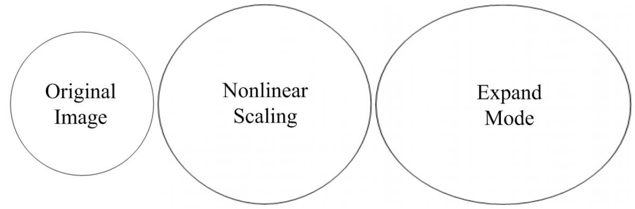


Fig.6 Comparison of nonlinear scaling and expand mode

IV. CONCLUTION

In this paper, a novel nonlinear scaling approach is proposed. Based on human visual theory, the approach first splited the input image into 3 parts, two nonlinear areas and a linear area, then applied different algorithms to different areas, the linear area was expanded through traditional algorithm, and the nonlinear areas were expanded by the proposed algorithm. Compared with other traditional approaches, the simulation results above shows that the proposed approach kept all the pixel data in the input video, also, the whole screen is crammed, the visual effect looks better than directly stretching the image to full screen.

ACKNOWLEDGEMENT

The authors acknowledge the support of the National Natural Science Foundation of China (60576025) and Tianjin Municipal Science and Technology Development Project (08ZCGYGX00400).

REFERENCES

[1] Hans Hoffmann, Takebumi Itagaki, "A Novel Method for Subjective Picture Quality Assessment and Further Studies of HDTV Formats", IEEE Transactions on Broadcasting, Volume 54, Issue 1, Page(s):1 – 13, 2008
[2] Ishii, Y, "The World of Liquid-Crystal Display TVs—Past, Present, and Future", Journal of Display Technology, Volume 3, Issue 4, Page(s):351 – 360, 2007
[3] Jun Xin, Ming-Ting Sun, Kangwook Chun, Byung Sun Choi, "Motion re-estimation for HDTV to SDTV transcoding", IEEE International Symposium on Circuits and Systems, Volume 4, Page(s):715 – 718, 2002
[4] Shih-Chang Hsia, Bin-Da Liu, "A parallel video converter for displaying 4:3 images on 16:9 HDTV receivers", IEEE Transactions on Circuits and Systems for Video Technology, Volume 6, Issue 6, Page(s):695 – 699,1996
[5] Zhongliang Deng, Yandong Guo, "A Comparative Review of Aspect Ratio Conversion Methods", International Conference on Multimedia and Ubiquitous Engineering, Page(s):114 – 117,2008
[6] Zhao Shuying, Yang Jinzhu, Chen Liying, "An Algorithm of Nonlinear Interpolation Segmentation Based on Distance Mask Applied to Human Brain 3D Reconstruction", 2nd International IEEE EMBS Conference on Neural Engineering, Page(s):37 – 40, 2005
[7] Nuno-Maganda, Arias-Estrada, "Real-time FPGA-based architecture for bicubic interpolation: an application for digital image scaling", International Conference on Reconfigurable Computing and FPGAs, Page(s):8, 2005
[8] Galdino, Borges, "A visual attention model for tracking regions based on color correlograms", Proceedings XIII Brazilian Symposium on Computer Graphics and Image Processing, Page(s):36 – 43, 2000