Precise Image Resampling Algorithm

Pavel Zemčík
Faculty of Information Technology
Brno University of Technology
Božetěchova 2
CZ 612 66, Brno, Czech Republic
zemcik@fit.vutbr.cz

Bronislav Přibyl
Faculty of Information Technology
Brno University of Technology
Božetěchova 2
CZ 612 66, Brno, Czech Republic
xpriby12@stud.fit.vutbr.cz

Adam Herout
Faculty of Information Technology
Brno University of Technology
Božetěchova 2
CZ 612 66, Brno, Czech Republic
herout@stud.fit.vutbr.cz

ABSTRACT

This paper introduces a precise image resampling algorithm intended for corrections of image distortions caused by lenses or similar devices. The algorithm is designed for correction of small distortions in terms of pixel displacement but with high subpixel precision. The geometrical description of the correction is through bi-linear interpolation within each node of a square or rectangular mesh. The paper describes the algorithms itself, its features, implementation issues and data formats. Specifically discussed are the issues connected with programmable hardware (FPGA) implementation.

Keywords

Image resampling, subpixel resampling, lens distortion, FIR filter, bilinear interpolation.

1. INTRODUCTION

Image processing is one of the fields of computer science and applications that is developing very fast. The object of image processing is, of course, an image. Vast majority of image processing methods assumes that the image is a 2D signal represented through samples organized in a regular square or rectangular raster [For02a]. While the contemporary image acquisition devices and methods acquire images that relatively well fulfill the above assumption, in most cases, the images suffer from small geometrical imperfections caused e.g. by lenses used with the cameras that acquire the images.

The geometrical imperfections are in some cases not critical; however, many applications of image processing exist that suffer from the imperfections and where it is desirable to correct them. While the geometrical correction – calculation of new sample positions in the image – is relatively straightforward and can be e.g. performed through bi-linear interpolation within square or rectangular mesh, the problem remains how to get the new samples' values so that the signal properties of the image remain as much preserved as possible. Unfortunately, the nearest neighbor method, which completely destroys the image signal properties, and bi-linear or bi-cubic interpolation [Gal05a] which can be better but by far is not ideal, are traditionally used for this purpose. The main reason is that while the algorithms to preserve good signal properties, namely frequency spectrum, are known, they are often considered prohibitively computationally expensive. This paper proposes a method that is far better from the point of view of signal properties than the bi-linear or bi-cubic interpolation while still preserves relatively low computational requirements. The limitation of the proposed method, however, is that it is limited to the cases where the distortions do not involve significant angular or scale changes – the method merely assumes subpixel shift limited to several pixels displacement [For02a], [Gal05a].

2. IMAGE RESAMPLING

General image resampling problem is relatively straightforward mathematically – it is merely a problem of proper reconstruction of signal values in 2D space and proper application of sampling theorem. However, the efficient implementation of such resampling is still quite open problem. In our approach, we limit the general problem to resampling in order to correct geometrical imperfections only. This limitation has the following implications:

- The displacement of pixel location of the original and resampled images is only units of pixels,
- no angular distortion is expected, and
- no scaling is expected.

The general approach for resampling in our case is to scan the output image raster pixel by pixel (sample by sample) and reconstruct the values from the original

raster based on knowledge of the pixel displacement. Taking into account the above limitations, it is known that the sampling theorem cannot be violated and also it can be assumed that the function is separable.

(1) $r_{x,y} = f(o, d(x,y)) = f''(f'(o, d_x(x,y)), (x,y))$

where ***r*** is the resampled image,

o is the original image,

d is the displacement function,

f is a resampling function,

and ***f'*** and ***f''*** are the partial reconstruction functions after separation.

In our case, the functions ***f'*** and ***f''*** are implemented through a bank of FIR filters indexed by subpixel location of the pixel. Moreover, the sampling is the same in both directions, so ***f'*** and ***f''*** are implemented using the same FIR bank.

The above solution with FIR filters was chosen as it has well defined features and as it is quite flexible in terms of exchangeability of the filtering function.

3. PROPOSED RESAMPLING

The proposed approach to resampling relies on the separability; however, in addition to the generally used approach, the separability is applied to both the resampling function itself and the geometrical distortion calculation so that the distortion calculation is separated in vertical and horizontal directions.

(2) $r_{x,y} = f''(f'(o, d_x(x,y)), d_y(x,y))$

where **r** is the resampled image,

o is the original image,

$\boldsymbol{d_x}$ and $\boldsymbol{d_y}$ are the displacement functions,

and ***f'*** and ***f''*** are the partial reconstruction functions after separation.

The resampling function itself is assumed to be some suitable filter function and in the presented approach it is implemented through a bank of FIR (Finite Impulse Response) filters [Rab78a]. The bank of FIR filters is indexed through a subpixel position of the desired sample in the raster. The reason is that the FIR coefficients are dependent on the subpixel position of the desired sample location. Of course, the size of FIR filters is limited. The filters in the bank can be e.g. Lanczos filters [Theu00a] for optimal exploitation of the bandwidth of the image signal given the size of the filter, or other filter design to achieve the desired image signal properties. The described approach is, in fact, not dependent on it.

(3) $r_{x,y} = FIR_{fp(x)}(FIR_{fp(y)}(o, ip(d_y(x,y))), ip(d_x(x,y)))$

where ***r*** is the resampled image,

o is the original image,

$\boldsymbol{d_x}$ and $\boldsymbol{d_y}$ are the displacement functions,

$\boldsymbol{FIR_t}$ is the function of the bank for position ***t***,

fp is the fractional part of a numerical value,

and ***ip*** is the integer part of a numerical value.

The distortion to be corrected is described with a square mesh with displacement specified for each node of the mesh. While the displacement in each node (corner of the squares) of that mesh is known, the displacement inside the squares is computed via bi-linear interpolation.

Distortion inside each square is described by means of the following four pre-calculated coefficients:

- D_0 – top left pixel displacement.
- DC_0 – difference of displacements between adjacent pixels in 1st row of the square.
- DR – difference of displacements between 1st pixels of adjacent rows.
- DDC – change in difference of displacements between pixels of adjacent rows, that means $DC_{n+1} - DC_n$.

For more detailed description see Figure 1. Note, please, that the displacement calculation can be subdivided into independent calculation of vertical and horizontal displacements.

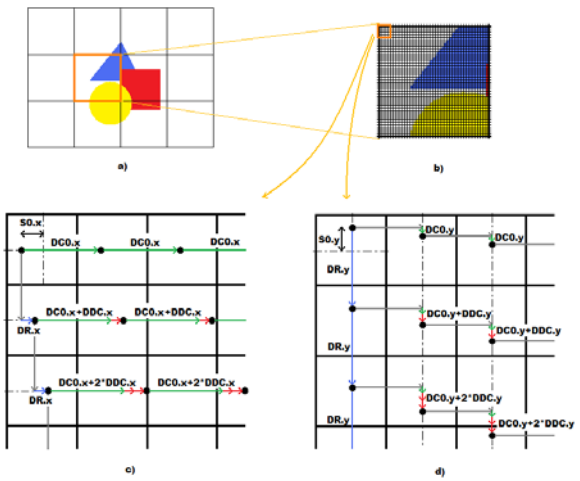

Figure 1: Displacement interpolation in squares

The following pseudocode illustrates displacement calculation resampling algorithm executed in each

square of the mesh. The input of the algorithm is the original image, distortion description (through the above mentioned coefficients), and FIR filter; the output is the resampled pixels within the given square. Note, please, that two instances of the algorithm are being used, one for vertical and one for horizontal displacement and filtering.

```
var DoR, DC, D;
DoR = D0;
DC = DC0;
(foreach row in square)
{
  D = DoR;
  (foreach pixel in row)
  {
    Output FIR[fp(D)](O,ip(D));
    D += DC;
  }
  DoR += DR;
  DC += DDC;
}
```

As it can be seen from the pseudocode, three variables are needed in the algorithm. Their meaning is as following:

- DoR – displacement of 1st pixel in a row.
- DC – difference of pixel displacements.
- D – displacement of current pixel.

As shown in the algorithm, the displacement is subdivided into integer and fraction parts. The integer part is used to determine the pixel placement of the filter while the fractional part is used to determine the set of coefficients within the filter bank. When the number of filters in the bank is N (e.g. 16), the fractional part is multiplied by N and then rounded to nearest integer. Then it is used for filter bank index.

4. FPGA IMPLEMENTATION

An FPGA [Bro96a] implementation of the resampling algorithm has been prepared as part of the experiments with the design. The dataflow in the resampling unit can be seen in Figure 2. The processing contains four parts which are associated in two groups. One group handles vertical resampling while the other handles horizontal resampling. Each group consists of a FIR module and Displacement interpolation module.

Data formats used in the algorithm are the fixed decimal point numbers in order to represent the data accurately enough while maintaining the design simple to enable its simple implementation.

The actual data formats used in the experiments are shown below.

- Pixel data – 16 bit signed or unsigned. The pixel processing is assumed in 16 bit format in order to support the standard dynamic range of contemporary video cameras, which is 10 to 14 bits, plus an overhead for absorption of rounding errors of FIR.
- Co-ordinate – 12+4 bits unsigned. The subpixel resolution is assumed to be 16 subpixel positions which is in practical terms enough to avoid measurable adverse effects of granularity in subpixel position.
- Difference of co-ordinates – 2+8 bits signed. The difference of positions must be precise enough to represent the change of displacement.

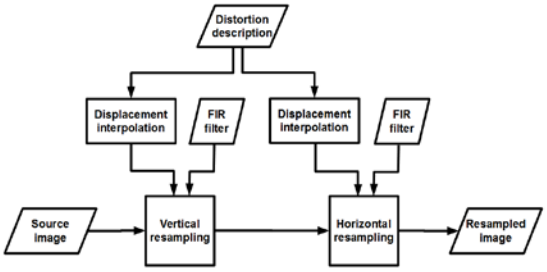

Figure 2: Dataflow of resampling algorithm.

The experimental design and synthesis of the resampling unit was performed for Xilinx Virtex-II xc2v1000 FPGA device. XST version H.38 was chosen for this task. As the unit is relatively generic, the following parameters were used: Image size 256 x 1024 px and square size 64 px which results in square mesh of 4 x 16 squares (and also displacement coefficient sets). Device utilization with configuration mentioned above is shown in Table 1.

Items on chip	Used	Capacity	% capacity
Slices	3 947	5 120	77%
Slice Flip Flops	2 212	10 240	21%
4 input LUTs	3 103	10 240	30%
BRAMs	20	40	50%

Table 1: Exploitation of FPGA unit Virtex II-1000

The device clock frequency is up to 105 MHz. While the resampling unit produces one output pixel per 2 clock cycles, the output resampling data rate for a single unit is up to 52.5 Mpixels per second. This demonstrates the real-time potential of the design.

5. RESULTS

The algorithm has been evaluated with images of cells obtained through microscopy, synthetic image with various shapes, and other images not shown with results identical for standard CPU implementation (in C) and FPGA implementation (in VHDL).

All the images were resampled using a geometrical correction of some lens imperfections. Along with the images themselves, their energy spectrum is shown to demonstrate very little loss of energy caused with the actual resampling. The resampling itself was performed with 7-sample Lanszos filter and the subpixel resolution was 16. These values are the limit values for the current implementation. These values can be seen as limits for efficient exploitation in real-life applications; however, they do not represent any limit for FPGA hardware. See Figure 3 and Figure 4 for examples of the algorithm results.

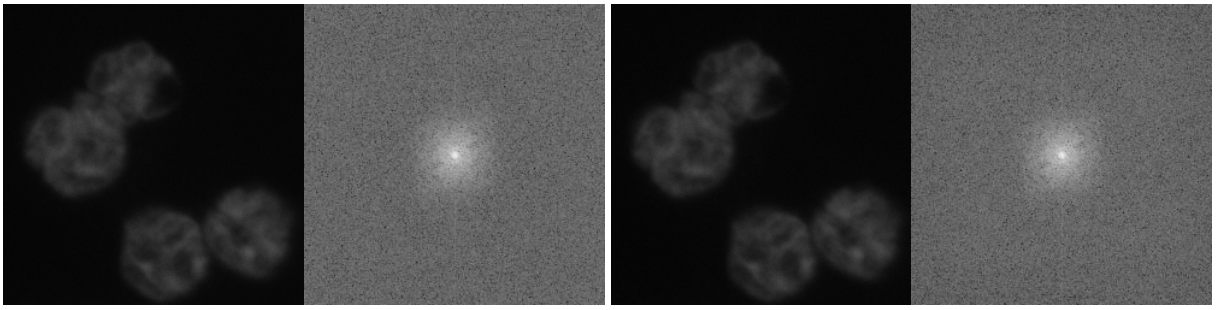

Figure 3: Cells – original image and its spectrum (left), resampled image and its spectrum (right)

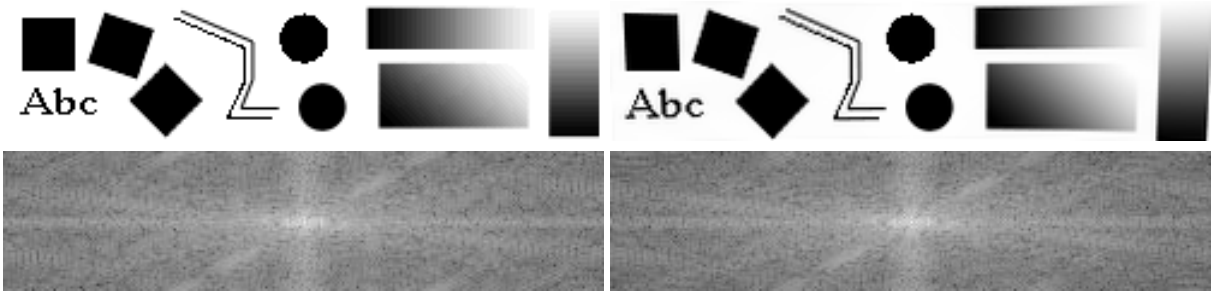

Figure 4: Shapes – original image and its spectrum (left), resampled image and its spectrum (right)

6. CONCLUSIONS

The goal of the contribution was to present a new resampling algorithm that is intended for corrections of geometrical distortions caused during image acquisition.

A new algorithm has been presented that exploits separable FIR filters and also separable displacement calculation for vertical and horizontal directions while it does not adversely affect the image.

The algorithm has been implemented and prepared also for FPGA exploitations. Using the Lanczos filter, the algorithm has also very good results in terms of quality of image signal. Future work should include further simplification of the algorithm.

7. ACKNOWLEDGMENTS

This work was supported by the grant project of the Ministry of Educatuon, Youth and Sports of CR, (MSMT 2B06052) project “BioMarker”.

8. REFERENCES

[Bro96a] Brown, S. and Rose, J. FPGA and CPLD architectures: A tutorial. IEEE DESIGN and TEST OF COMPUTERS, pp. 42-57, 1996.

[For02a] Forsyth, D. A. and Ponce, J. Computer vision: A modern approach. Prentice Hall Professional Technical Reference, New Jersey, 2002.

[Gal05a] Gallagher, AC. Detection of linear and cubic interpolation in JPEG compressed images. In proceedings of The 2nd Canadian Conference on Computer and Robot Vision, pp. 65-72, Victoria, BC, Canada, 2005.

[Rab78a] Rabiner, LR and Gold, B. and Yuen, CK. Theory and application of digital signal processing. IEEE Transactions on Systems, Man and Cybernetics, vol. 8, nr. 2, pp. 146, 1978.

[Theu00a] Theußl, T. and Hauser, H. and Gröller, E. Mastering windows: Improving reconstruction. In Proceedings of the 2000 IEEE symposium on Volume visualization, pp. 101-108, ACM New York, NY, USA, 2000.