REAL-TIME KEYSTONE CORRECTION OF VIDEO IMAGE USING FPGA

Zmago Jereb[1], Janez Diaci[2]

[1] Kolektor Group d.o.o., Vojkova 10, 5280 Idrija, Slovenia
[2] University of Ljubljana, Faculty of Mechanical Engineering, Ljubljana, Slovenia

Key words: real-time image processing; keystone deformation; FPGA

Abstract: In this paper we present a method for geometrical keystone correction of a video image using digital image processing. The method is divided into image deformation analysis and real-time image correction. The analysis is based on comparing a projected reference image against its original. The transformation is formulated as a sequence of simple operations which can be implemented for real-time execution using existing hardware technology. An implementation uses a programmable gate array (FPGA) and demonstrates the feasibility of real-time transformation of the image. The aim of the implementation is to minimize the hardware footprint and allow using a low-cost FPGA device.

Digitalna geometrijska korekcija trapeznega popačenja projicirane video slike v realnem času z uporabo FPGA

Kjučne besede: realni čas; obdelava slik; trapezna deformacija; FPGA

Izvleček: To delo podaja metodo, ki omogoča digitalno geometrijsko korekcijo trapeznega popačenja projicirane video slike. Metoda je razdeljena na analizo popačenja prikazane slike in njeno korekcijo v realnem času. Analiza deformacije je osnovana na podlagi referenčne slike, katere original je primerjan s projicirano sliko. Digitalna korekcija slike zajema zaporedje matematičnih operacij, ki so zasnovane z namenom implementacije v obstoječe sisteme za obdelavo slik v realnem času. V delu je predstavljena implementacija z uporabo integriranega vezja s poljem logičnih vrat (FPGA), ki prikazuje zmožnosti metode. Implementacija je izvedena s poudarkom na minimizaciji potrebnih logičnih funkcij, s čimer je dana možnost uporabe manjših in cenejših FPGA integriranih vezij.

1. Introduction

Keystone deformation is a common geometric distortion of a projected video image. A typical installation of a video projector under the ceiling, aimed with its optical axis at an arbitrary vertical angle to the projection screen, causes a vertical keystone deformation of the projected image. Common video projection systems incorporate build-in correction based on a lens displacement /1/, /2/. The alternative to the optical correction method is the digital image correction method, which remaps the image pixels of the original image to the pre-warped image in such a way, that the projected image appears undistorted.

The available literature addresses the geometrical transformations with a use of graphic card libraries (e.g. OpenGL) /3/, /4/ or with special hardware designs.

The designs using a field programmable gate array (FPGA) devices have become a popular practice in real-time image processing studies. The reprogrammable logic and the capability to implement parallel processing make FPGA applicable for real-time image processing /5/, /6/, /7/, /8/, /9/. The methods often require relatively large silicon footprint and external high-bandwidth data buffers due to large amount of image data that needs to be processed in a short period of time /9/.

In this paper we present an FPGA based digital image processing method that allows a vertical keystone correction of the projected video image in real-time. The implementation is designed with a consideration to minimize the FPGA resources and the other supplementary hardware to achieve a low-cost solution. The algorithm is based on a comparison between the reference points taken from the original image and the points projected on the projection screen. The presented experiments demonstrate the potential of the method to remove the keystone image deformation as well as more complex horizontally warped image deformation.

2. Method

The presented method is shown schematically in Fig. 1. The basic video projection set-up: a multimedia player, a video projector and a projection screen is augmented by three additional components: a camera, an image comparator and an image processor. The image processor, placed between the multimedia player and projector, receives the original video image I_{in}, geometrically corrects it in real-time and sends the corrected image I_{out} to the projector.

The function of the camera and the image comparator is to determine the parameters c_{ik} of the correction algorithm. The transformation addresses vertical keystone distortions

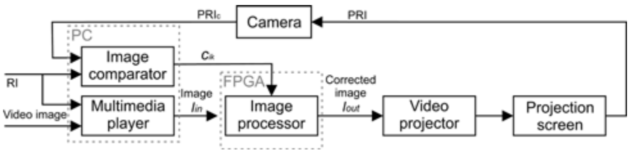

Fig. 1: Schematic representation of the method.

and is therefore applied to the image in the horizontal direction only.

The reference image (RI) and its projected image (PRI) represent an orthogonal lattice of nine equally distributed horizontal and vertical lines as illustrated in Fig.2. The intersections between the two outer left and right vertical lines and nine horizontal lines represent the coordinates (x_{ij}, y_{ij}) and (u_{ij}, v_{ij}) for the RI and PRI image respectively.

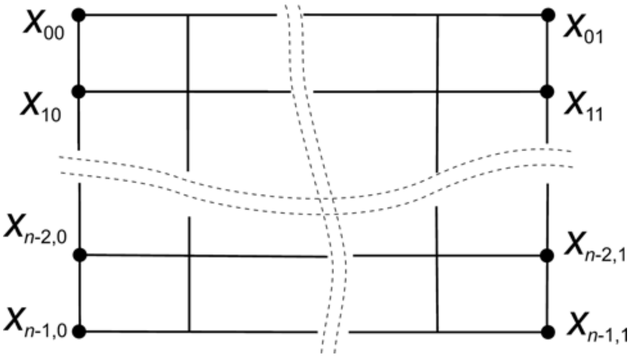

Fig. 2: Reference points on the original image.

The coordinate systems of (x,y) and (u,v), in general, do not match. Therefore, the u_{ij} values are translated and scaled to match the x_{ij} counterparts before the calculation of the c_{ik} parameters. The following equations are presented assuming the RI image of size 1024 x 768 pixels (XGA).

$$u_{ij}^{'} = \left(u_{ij} - \max_{0\le l\le 8}(u_{l0})\right) \Big/ \left(\min_{0\le l\le 8}(u_{l1}) - \max_{0\le l\le 8}(u_{l0})\right)\cdot 1023 \qquad (1)$$

To simplify the notation we use the symbols of the calibrated values u'_{ij} from Eq. 1 as u_{ij} from here on.

The transformation functions $u(x,y)$ and $v(x,y)$ are determined as a collection of bi-linear functions $u_i(x,y)$ and $v_i(x,y)$, each defined on a domain bound by four neighboring reference points: $x_{i0} < x < x_{i,1}$ and $y_i < y < y_{i+1,j}$. The horizontal functions are defined as:

$$u_i(x,y) = \begin{bmatrix} c_{i0} & c_{i1} & c_{i2} & c_{i3} \end{bmatrix} \begin{bmatrix} 1 \\ x \\ y - y_i \\ x(y - y_i) \end{bmatrix} \qquad (2)$$

where the parameters c_{ik} are defined from the intersection coordinates x_{ij}, y_{ij} and their projected counterparts u_{ij} :

$$\begin{aligned} c_{i0} &= u_{ij} \\ c_{i1} &= \left(u_{i,1} - u_{i0}\right) \big/ \left(x_{i,1} - x_{i0}\right) \\ c_{i2} &= \left(u_{i+1,0} - u_{i0}\right) \big/ \left(y_{i+1,0} - y_{i0}\right) \\ c_{i3} &= \left(u_{i0} - u_{i+1,0} - u_{i1} + u_{i+1,1}\right) \big/ \left(\left(x_{i,1} - x_{i0}\right)\left(y_{i+1,0} - y_{i0}\right)\right) \end{aligned} \qquad (3)$$

The vertical transformation functions are defied as identity transformations in accord with the above assumption of vertical keystone deformation:

$$v_i(x,y) = y + y_i \qquad (4)$$

In the final corrective transformation of the method, the pixel intensity values of the input image are remapped using the inverse mapping algorithm /10/ that remaps the pixel intensity values of I_{in} to the corresponding pixel intensity values of I_{out}:

$$I_{out}[x,y] = I_{in}[\lfloor u \rfloor, y] + \{x\}\left(I_{in}[\lceil u \rceil, y] - I_{in}[\lfloor u \rfloor, y]\right) \qquad (5)$$

where $\{x\} = x - \lfloor x \rfloor$, $\lfloor x \rfloor$, and $\lceil x \rceil$ represent the fractional part, floor, and ceiling functions. The u and v are shorthand notations for the transformation functions of $u(x,y)$ and $v(x,y)$, respectively.

The above transformation also employs image filtering based on linear interpolation between neighboring pixel intensity values. The filter improves the quality of the transformed image by reducing the aliasing effect of the transformation. The filter is applied to each color channel (red, green, blue) of the image separately (I_{outr}, I_{outg}, I_{outb}).

3. Implementation

A schematic organization of the method implementation is shown in Fig. 3. The implementation is designed to apply horizontal geometrical correction to a 24 bit XGA video image (1024 pixels wide, 768 pixels tall) at the 60 fps rate. The transformation parameters c_{ik} are calculated with a personal computer using Eq. (3) and transmitted to the image processor over a RS232 serial link. The digital video (DVI) serial protocol is handled by two interface circuits (Texas instruments TFP101 and TFP410) which transform fast serial DVI data to parallel data and vice versa. The image processor algorithm is implemented using *Digilent Inc. Spartan-3 starter board*, based on Xilinx XC3S200 FPGA. The FPGA algorithms are developed with Xilinx ISE development environment and the VHDL hardware description language.

The video image data is transformed line by line starting from the top-left corner of the image. The presented correction is applied to the image horizontally, so that it coincides with the image data flow. For that reason, only two image lines of data are stored at a time. While buffer1 receives a new image line, the line buffer2 provides the previous image line to the image correction block. When the

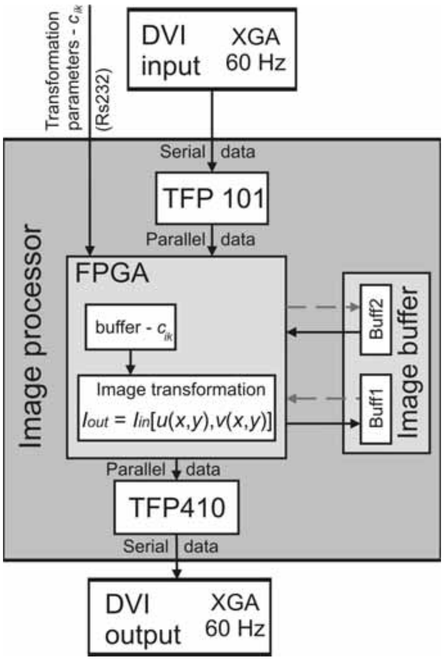

Fig. 3: Image processor block diagram.

buffer1 gets full and simultaneously the image transformation is complete, the two buffers swap their functions.

The calculation of the transform function (Eq. (2)) is implemented as a series of additions as shown in Eq. (6) where the variable Δy represents the term y-y_i. For the terms that use a multiplication, the linearity is exploited with the result that only an addition is needed when the x or Δy are incremented. The variables and the result are truncated to the most significant 24 bits.

$$u_i(x,y) = c_{i0} + \left(c_{i1} + \left(c_{i3}\Delta y\right)\right)x + c_{i2}\Delta y \qquad (6)$$

The remapping algorithm with bi-linear anti-aliasing filter (Eq. (5)) requires one hardware multiplier for each color. The filter is calculated using two neighboring pixel intensity values floor(u) and ceiling(u) from the image line buffer. The image pixels are stored in the buffer in pairs (...; $\{I_{in}(x_i),I_{in}(x_{i+1})\}$; $\{ I_{in}(x_{i+1}),I_{in}(x_{i+2})\}$; ...). The reason for doubling the data is to simplify the calculation and speed-up the transformation process. The above solution uses only one access to buffer to provide the data needed to calculate the interpolated value for the I_{out}.

The result of the transformation function u_i (Eq. (6)) may fall out of the image range $0<u(x,y)>1024$. For these pixels, Eq. 5 is bypassed and the intensity value is set to black color ($I_{out\,r,g,b}(x,y) = (0, 0, 0)$) which represents the color of the background.

The transformation process is synchronized with the input image data flow and has a latency of one image line. The presented FPGA implementation uses 455 slices, three 18 x 18 bit hardware multipliers, 7 Block RAMs (126 kb) and 58 I/O pins.

4. Results and discussion

Experimental set-up used to validate the method and implementation includes a video projector Hitachi-ED-X12, a digital camera Canon EOS 350D with EF-S 18-55 mm lens, a personal computer and the image processor.

The obtained results are illustrated in Fig. 4. In the first experiment we tested a keystone deformation. The deformed PRI and corrected PRI images are shown on Figs. 4a and 4b respectively. In the second experiment illustrated in Fig 4-c,d, we tested the ability of the method to correct warped image deformations. The warped deformation was obtained by a small warp on the bottom of the projection screen.

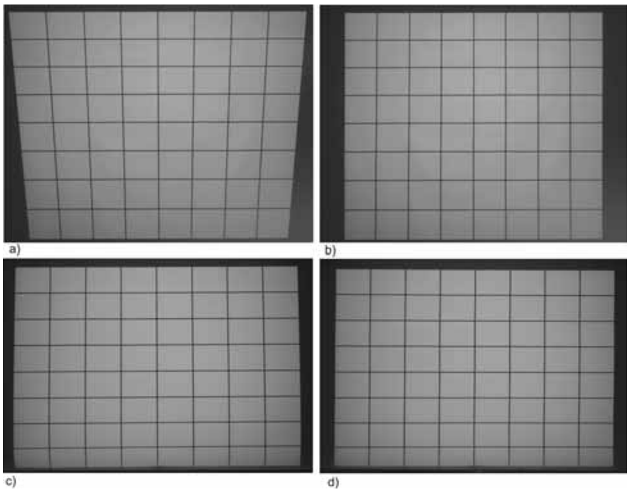

Fig. 4: Keystone deformed (a) and corrected image (b) and warped screen deformation (c) and its corrected image (d).

The measured error values are presented in Table 1 which shows the displacements of the reference points obtained from the deformed image and the residual displacements after applying the correction.

In both cases, the results show a great reduction of error after applying the correction. The residual error of the keystone and warped image correction doesn't exceed 2 pixels and 4 pixels respectively.

The results demonstrate that the method enables a significant reduction of the horizontal image distortion in real-time. The method is therefore applicable to the correction of the vertical keystone deformations frequently present in the projection systems as well as mild horizontally warped images which can be addressed due to the multiple reference points along the vertical image borders.

The drawback of the presented method is the change of aspect ratio due to horizontal shrinkage of the corrected image. The method is therefore limited to mild deformations which don't noticeably change the aspect ratio.

Table 1: Reference point displacements.

	Keystone [pix]				Warped image [pix]			
	original		corrected		original		corrected	
i	x_{i0}	x_{i1}	u_{i0}	u_{i1}	x_{i0}	x_{i1}	u_{i0}	u_{i1}
0	72	64	0	0	0	0	2	0
1	63	57	0	0	1	3	2	0
2	53	49	0	0	3	5	3	0
3	44	40	0	2	4	7	2	1
4	36	33	0	0	6	9	2	2
5	27	25	0	0	6	11	2	2
6	18	16	0	0	7	12	2	3
7	9	8	1	0	7	12	0	4
8	0	0	1	0	0	8	0	2

With expansion of the reference points lattice over the hole image, the method could be also used to address a specific horizontal image deformations introduced by an omnidirectional display /11/, /12/.

The presented work is focused on real-time image transformation whereas the deformation analysis is performed off-line. Our future work will focus on development of real-time image deformation detection, which would allow on-line adaptation of image correction parameters.

5. Conclusion

We present a method for real-time horizontal geometrical distortion of video image based on digital image processing. The algorithms are divided into image deformation analysis implemented on a personal computer and real-time image processing implemented with a FPGA. The presented implementation allows the use of a small and low cost FPGA device.

The experimental validation of the method has been performed on a keystone deformed image and on a warped image. The results show that the method can significantly improve the geometry of an image.

The limitation of the proposed method is horizontal shrinkage of the correction image that results in a change of the aspect ratio. Therefore, for severely distorted image it would be more appropriate to apply two dimensional correction.

6. References

/1/ A. Stolov, "Projector system including keystone correction", U.S. Patent 5706062, 1998.

/2/ J.R. Biles, G.B. Kingsley, A.R. Conner, "Method and apparatus for distortion correction in optical projectors", U.S. Patent 5355188, 1994.

/3/ M. Brown, W. Seales, "A Practical and Flexible Tiled Display System", Proceedings of the 10th Pacific Conference on Computer Graphics and Applications, 194 - 203, Oct. 2002.

/4/ J.P. Tardif, S.Roy, M. Trudeau, "Multi-projectors for arbitrary surfaces without explicit calibration nor reconstruction", Proceedings of the Fourth International Conference on 3-D Digital Imaging and Modeling, 217 – 224, 2003.

/5/ N. Sedcole, P. Cheung, G. Constantinides, W. Luk, "A Reconfigurable Platform for Real-Time Embedded Video Image Processing", Field-Programmable Logic and Applications, 606-615, 2003.

/6/ P. Sedcole, B. Blodget, T. Becker, J. Anderson, P. Lysaght, "Modular dynamic reconfiguration in Virtex FPGAs", IEE Proceedings on Computers and Digital Techniques, vol. 153, 157-164, 2006.

/7/ F. Kopač, A. Trost, "A systematic approach to real-time image segmentation in FPGA devices", Inf. MIDEM, vol. 35, No. 1, 13-19, 2005.

/8/ A. Žemva, A. Trost, B. Zajc, "Educational programmable system for prototyping digital circuits", Int. J. of Electrical Engineering Education, 1998, Vol. 35, No. 3, pp. 236-244.

/9/ D. Eadie, F.P. Shevlin, A. Nisbet, "Correction of geometric image distortion using FPGAs", Proceedings of Opto-Ireland 2002: optical metrology, imaging and machine vision, 28-37, June 2003.

/10/ K.R. Castleman, "Digital image processing", Prentice Hall, 1996.

/11/ D. Jurjavčič, "Device providing picture visibility from all sides," U.S. Patent 6460278, 2002.

/12/ J. Babic, "Device providing simultaneous visibility of images within the area of 360 around itself," U.S. Patent 20060179693, 2003.

Zmago Jereb
Kolektor Group d.o.o.
Vojkova 10, 5280 Idrija, Slovenia
zmago.jereb@kolektor.si

Janez Diaci
University of Ljubljana, Faculty of Mechanical
Engineering, Aškerčeva 6, Ljubljana, Slovenia

Prispelo (Arrived): 17.01.2010 Sprejeto (Accepted): 09.09.2010