ITG Fachbericht 167 - Elektronische Medien, Systeme, Anwendungen,
VDE Verlag, Berlin, pp. 281-286, 2001

VLSI Architecture for Electronic Correction of Optical Distortions

Marco Winzker, Uwe Rabeler
Liesegang electronics GmbH, Hannover, Germany

Abstract

Optical distortions can occur for image acquisition as well as for image presentation. Sources for these distortions are presented and their characteristics are analyzed. Based on the extracted characteristics a VLSI architecture for electronic correction of optical distortions is proposed.

1. Introduction

For image acquisition with an electronic camera a scene in the real world is projected by a complex lens system on an image sensor and transferred to an electronic signal. For image presentation with a projector, the electronic image information is presented on a digital display device, e.g. an LCD or DMD. Again a lens system is used to project the image onto a screen or directly on the retina of a person's eye.

In the camera as well as the projector an optical system and an electronic system work closely together. Consequently, during the design of the electronic system the characteristics of the optical system and vice versa have to be considered. Furthermore, one system can compensate the deficiencies of the other. Due to the advances in VLSI technology leading to increasing computing power for decreasing cost, it is the electronic system that can compensate more and more optical distortions.

Optical distortions have been accepted for some applications because they need only qualitative measurement and do not require geometric accuracy. For other applications, e.g. image projection, geometric accuracy is not absolutely required but becomes a quality feature. A further range of applications really requires geometric accuracy.

In this paper we present a VLSI architecture for correcting a wide range of geometric distortions. The next chapter will present some sources of optical distortion and thus define the requirements for electronic correction. In chapter three we present an architecture for electronic distortion correction and chapter four discusses an ASIC implementation currently under development.

2. Sources of Optical Distortion

Notation: Undistorted coordinates will be nominated as (x, y) and distorted coordinates as (x', y'). Origin is the optical axis.

Optical distortions have to be considered for image acquisition as well as for image presentation. For better readability, we will refer in the remainder of this text only to projection systems. Nevertheless, the results are also applicable for image acquisition.

2.1. Lens Distortion

Real lens systems are non-linear and introduce a radial shift to the position of the image pixel.

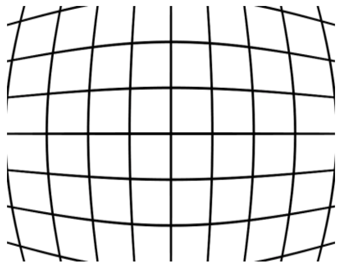

Fig. 1: Barrel distortion of a rectangular grid of lines

This shift can result in a radial distortion called pincushion or barrel distortion. Figure 1 shows the effect a barrel distortion has on a rectangular grid of lines.

Radial distortion can be described as a coordinate transformation [1]. The image information from position (x, y) is transformed to position (x', y') according to:

$$x' = x + k_1 \bar{x} d^2 + k_2 \bar{x} d^4 + k_3 \bar{x} d^6 + P_1 (2\bar{x}^2 + d^2) + 2 P_2 \bar{x} \bar{y} \quad (1)$$

$$y' = y + k_1 \bar{y} d^2 + k_2 \bar{y} d^4 + k_3 \bar{y} d^6 + P_2 (2\bar{y}^2 + d^2) + 2 P_1 \bar{x} \bar{y} \quad (2)$$

with $\bar{x}=x-u_0$, $\bar{y}=y-v_0$, $d^2=\bar{x}^2+\bar{y}^2$ and (u_0, v_0) as the optical center of the image, also denoted as the optical axis.

The first order distortion, denoted as k_1, accounts for the mayor part of the distortion. A simplified model of barrel distortion disregards the other coefficients and is written as

$$x_b' = x + k_1 \bar{x} d^2 \tag{3}$$

$$y_b' = y + k_1 \bar{y} d^2 \tag{4}$$

2.2. Rotation

A rotation between the display device and the projection screen can be expressed by multiplying the original coordinates with the rotation matrix R

$$R = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix} \tag{5}$$

giving

$$x_r' = x\cos\alpha - y\sin\alpha \tag{6}$$

$$y_r' = x\sin\alpha + y\cos\alpha \tag{7}$$

2.3. Projection Systems

A projection system normally is designed to project the image of a display device on a vertical projection screen. As a projector is usually placed on a desk and projects slightly upwards, a certain projection angle is often considered, e.g. by shifting the display device from the center of the optical axis. This is illustrated by Figure 2.

Assuming undistorted projection, a picture element (x_d, y_d) of the display device is projected to a position on the screen (x_p, y_p), which relate by a magnification factor *m*.

$$(x_p, y_p) = (m\cdot x_d, m\cdot y_d) \tag{8}$$

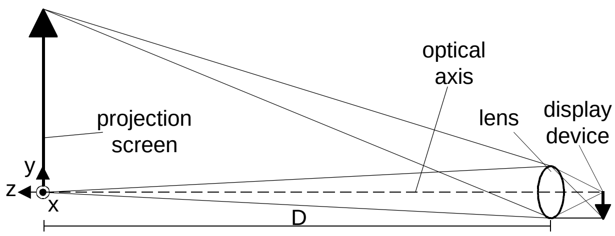

Fig. 2: Projection system

According to Figure 2, we define a coordinate system x,y,z. The projection screen $\bar{\bar{P}}$ is located in the origin and spans along the x- and y-coordinates.

$$\bar{\bar{P}} = \begin{pmatrix} u \\ v \\ 0 \end{pmatrix} \tag{9}$$

For typical projection applications, the dimensions of the lens are much smaller than the projection distance D. Then, the light source can be assumed to be a single point. In Figure 2, the projection lens is located at position $-D$ on the z-axis. A ray of light from the lens at position $(0,0,-D)$ towards the projected image pixel at $(x_p, y_p, 0)$ is defined as

$$\vec{r} = \begin{pmatrix} r\cdot x_p \\ r\cdot y_p \\ D(r-1) \end{pmatrix} \tag{10}$$

2.4. One-Dimensional Keystone Distortion

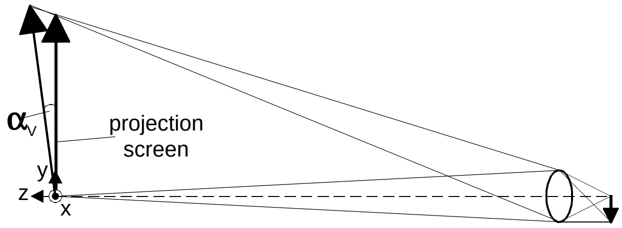

Fig. 3: Tilted projection system

When the angle of projection is different from the default angle, the optical system behaves as if it projects on a tilted screen (see Figure 3). The rays towards the top of the screen have a larger distance from the lens and the projected image becomes wider at the top. This so called keystone distortion results in a trapezoid-like image as depicted in Figure 4.

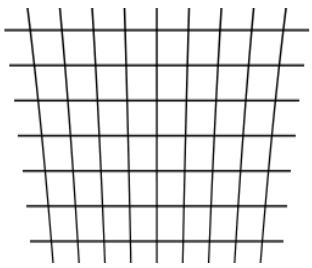

Fig. 4: Keystone distortion of a rectangular grid of lines

A projection screen, tilted from the y-axis by an angle of α_v (compare Fig. 3) is described by multiplying $\bar{\bar{P}}$ (equation 9) with the rotation matrix

$$R_v = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha_v & -\sin\alpha_v \\ 0 & \sin\alpha_v & \cos\alpha_v \end{pmatrix} \tag{11}$$

giving

$$\bar{\bar{P}}' = R_v \cdot \bar{\bar{P}} = \begin{pmatrix} u \\ v\cos\alpha_v \\ v\sin\alpha_v \end{pmatrix} \tag{12}$$

The intersection of the projection rays $\vec{r}$ towards the undistorted coordinates (x_p, y_p) with the tilted screen $\bar{\bar{P}}'$ has to be determined to calculate the distorted coordinates (x_k, y_k). Setting $\bar{\bar{P}}' = \vec{r}$ gives

$$p' = \begin{pmatrix} x_k \\ y_k \\ z_k \end{pmatrix} = \begin{pmatrix} \dfrac{x_p}{1 - \dfrac{y_p}{D}\tan\alpha_v} \\ \dfrac{y_p}{1 - \dfrac{y_p}{D}\tan\alpha_v} \\ \dfrac{y_p \tan\alpha_v}{1 - \dfrac{y_p}{D}\tan\alpha_v} \end{pmatrix} \tag{13}$$

The coordinates of p' have to be transformed from the original coordinate system to the coordinate system of the tilted screen by multiplying with the rotation matrix (equation 11). Thus, the keystone distorted image is defined as

$$p'_k = \begin{pmatrix} x'_k \\ y'_k \\ 0 \end{pmatrix} = R \cdot p' = \begin{pmatrix} \dfrac{x_p \cos a_v}{\cos a_v - \dfrac{y_p}{D}\sin a_v} \\ \dfrac{y_p}{\cos a_v - \dfrac{y_p}{D}\sin a_v} \\ 0 \end{pmatrix} \tag{14}$$

2.5. Two-Dimensional Keystone Distortion

When the projection is performed upwards and from a side, the screen is tilted in vertical and horizontal direction relative to the projector.

Instead of one rotation relative to the x-axis with the matrix R_v, the screen is also rotated relative to the y-axis with

$$R_h = \begin{pmatrix} \cos\alpha_h & 0 & -\sin\alpha_h \\ 0 & 1 & 0 \\ \sin\alpha_h & 0 & \cos\alpha_h \end{pmatrix} \tag{15}$$

Then $\bar{\bar{P}}''$ is

$$\bar{\bar{P}}'' = R_h \cdot R_v \cdot \bar{\bar{P}} = \begin{pmatrix} u\cos\alpha_h - v\sin\alpha_v \sin\alpha_h \\ v\cos\alpha_v \\ u\sin\alpha_h + v\sin\alpha_v \cos\alpha_h \end{pmatrix} \tag{16}$$

and the intersection with $\vec{r}$ is

$$p'' = \begin{pmatrix} \dfrac{x_p}{1 - \dfrac{x_p}{D}\tan\alpha_h - \dfrac{y_p}{D}\dfrac{\tan\alpha_v}{\cos\alpha_h}} \\ \dfrac{y_p}{1 - \dfrac{x_p}{D}\tan\alpha_h - \dfrac{y_p}{D}\dfrac{\tan\alpha_v}{\cos\alpha_h}} \\ -D + \dfrac{D}{1 - \dfrac{x_p}{D}\tan\alpha_h - \dfrac{y_p}{D}\dfrac{\tan\alpha_v}{\cos\alpha_h}} \end{pmatrix} \tag{17}$$

Transforming the coordinates towards the tilted screen gives

$$p''_k = \begin{pmatrix} \dfrac{x_p \cos\alpha_v + y_p \sin\alpha_v \sin\alpha_h}{\cos\alpha_v \cos\alpha_h - \dfrac{x_p}{D}\cos\alpha_v \sin\alpha_h - \dfrac{y_p}{D}\sin\alpha_v} \\ \dfrac{y_p \cos\alpha_h}{\cos\alpha_v \cos\alpha_h - \dfrac{x_p}{D}\cos\alpha_v \sin\alpha_h - \dfrac{y_p}{D}\sin\alpha_v} \\ 0 \end{pmatrix} \tag{18}$$

2.6. Projection onto Curved Surfaces

With the same approach as above a projection on curved surface can be described and the resulting distortion be calculated. [2] describes an application for rear projection onto a translucent sphere.

2.7. Brightness Variation

Due to variations in the light source and the lens system the brightness of a projected image normally is non-uniform. The brightness of a projected image can be measured at 9 points according to Figure 5.

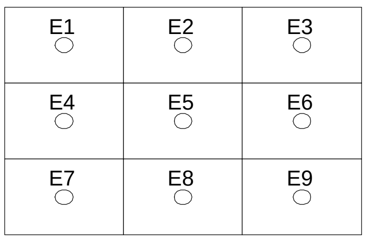

Fig. 5: Brightness measurement positions

Brightness uniformity G can be defined as the ratio between the brightest and darkest of these nine measurement positions. According to manufacturers data, the brightness uniformity of current current projectors is between 80% and 95% [3]. Measurements showed that brightness uniformity can be even lower than 60% [4].

While this uniformity only considers the nine measurement positions, the difference between the brightest and the darkest spot on the screen is usually even larger.

Optical distortion adds further brightness variation, as projection from an angle is less intensive as direct projection. The brightness distribution for distorted projection can be calculated as the intensity $I(x_{p,} y_p)$ for undistorted projection multiplied with the ratio between area in distorted $(\Delta x_k \cdot \Delta y_k)$ and undistorted $(\Delta x_p \cdot \Delta y_p)$ projection.

Thus, the relative brightness $i(x_{p,} y_p)$ is equal to the inverse of the derivative of the coordinate transformation between of the optical distortion.

$$i(x_p, y_p) = \frac{1}{\frac{d}{dx_p} x_k(x_p, y_p) \cdot \frac{d}{dy_p} y_k(x_p, y_p)} \quad (19)$$

For example, one-dimensional keystone correction according to equation (14) leads to a relative brightness distribution of

$$i_k(x_p, y_p) = \cos\alpha_v \left(1 - \frac{y_p}{D} \tan\alpha_v\right)^3 \quad (20)$$

3. System Architecture for Distortion Correction

Algorithms for distortion correction on general purpose computers have been discussed in literature [5]. Unless a very powerful computer is used, no real-time distortion correction is possible with this approach.

In this chapter, a VLSI architecture for correction of geometric and brightness distortions is described. Both corrections are performed independently from each other.

3.1. Correction of Geometric Distortions

For undistorted projection, the relation between a position (x_d, y_d) on the display device and the projected position is just a magnification (compare equation 2).

For distorted projection the image information that should be displayed at position (x_p, y_p) appears at position (x'_p, y'_d). To compensate this distortion, the transformation $(x_p, y_p) \rightarrow (x'_p, y'_d)$ must be known and considered for the display device.

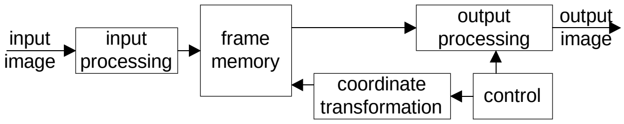

Fig. 6: Architecture for geometric correction

A circuit architecture for this compensation is depicted in Figure 6. First, the input image is stored in a frame memory, to allow random access to the image information. Digital output devices are usually addressed line-by-line. For each output pixel, a coordinate transformation calculates, where this pixel will be displayed on the output screen. Then the image information corresponding to the screen position is fetched from the frame memory and displayed on the output device.

3.2. Coordinate Transformation

The coordinate transformation must be able to cover the different distortion patterns or even combinations of two or more distortions discussed in Chapter 2.

For a given distortion, the trigonometric terms appearing in the equations of Chapter 2 are constant. Therefore, equation 18can be written as

$$p''_k = \begin{pmatrix} \frac{a_{01} x_p + a_{10} y_p}{b_{00} + b_{01} x_p + b_{10} y_p} \\ \frac{a'_{10} y_p}{b_{00} + b_{01} x_p + b_{10} y_p} \\ 0 \end{pmatrix} \quad (21)$$

with constant factors $a_{01,} a_{10,} b_{00,} \ldots$ depending on the angles α_h and α_v.

A generalized equation for coordinate transformation is the rational function

$$f(x,y) = \frac{a_{00} + a_{01} x + a_{10} y + a_{02} x^2 + a_{11} xy + a_{20} y^2 + \ldots}{b_{00} + b_{01} x + b_{10} y + b_{02} x^2 + b_{11} xy + b_{20} y^2 + \ldots} \quad (22)$$

However, due to the division, this function leads to a rather complex hardware implementation. Furthermore equation 22 is undefined at the roots of the denominator and can become discontinuous. As the image distortions discussed previously usually do result in steady functions with steady derivatives, equation 22 can be approximated by a polynomial function

$$f(x,y) = c_{00} + c_{01} x + c_{10} y + c_{02} x^2 + c_{11} xy + c_{20} y^2 + \ldots \quad (23)$$

This function can be calculated with lower hardware effort. In the proposed VLSI architecture, transformation of x- and y-coordinates are performed by independent function blocks.

3.3. Aliasing

In many cases, the coordinate transformation will not give the position of an existing input pixel but will result in a position between two pixel. The easiest approach for calculating an output pixel would be to fetch the nearest existing pixel. However, the results of this simple approach are not acceptable.

Bilinear filtering gives significantly better results, taking into account the neighbouring pixel and weighting them according to the sub-pixel position.

Optimal results can be achieved by anti-aliasing filters. This approach considers, that an image signal is a discrete-time signal, sampled with a certain (spatial) frequency. Scaling this signal is equivalent to resampling it with another frequency [6]. Two cases have to be distinguished: Upscaling, where the new frequency is higher than the original frequency and downscaling, where the new frequency is lower than the original frequency.

If a signal is upscaled, a higher frequency spectrum can be represented after resampling and the new frequency range is occupied by mirrored images of lower frequencies. This effect is known as 'aliasing'. However, as the mirrored signal components do not represent information from the input signal they should be removed by lowpass filtering to avoid visible artifacts.

For downscaling the input signal has to be lowpass filtered before resampling, to avoid aliasing when high frequency components of the input signal are mirrored to lower frequencies.

3.4. Memory Organization

For the frame memory a trade-off between price and memory access has to be found. To avoid so-called frame-tears, i.e. artifacts due to overlapping write and read pointers, two frames must be stored. Storing two UXGA images with 1200 lines, 1600 pixel per line and 24 bit per pixel requires a memory capacity of 88 Mbit.

This capacity can be provided by a single DDR-DRAM [7]. However, this memory does not allow a fast random access. Thus, an on-chip cache for the image data is required.

The output of the geometry correction is calculated line-by-line. For geometry correction, as discussed in Chapter 2, the transformed pixel coordinates (denoted as '×') follow a scan-line as shown in Figure 7(a). For most practical applications, the slope of this scanline is relatively flat, which can be used for the cache control. For calculation of an output pixel, the input data (denoted as 'o') in a window around the calculated output position are required (compare Figure 7(b)). The size of the window depends on the aliasing filtering. Typical values for n_H and n_V are between 2 (bilinear filtering) and 8 to 16.

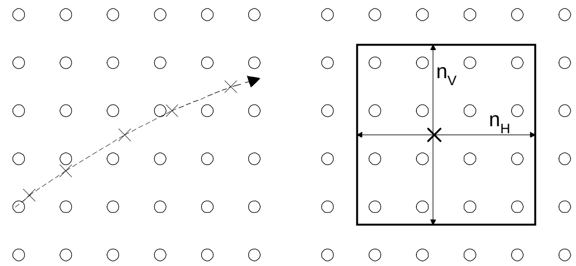

Fig. 7: (a) Scanline of output coordinates (×) (b) Window of required input pixel (o)

The cache control uses the result of the coordinate transformation and fetches segments of data around the scanline [8]. Unlike in [8], the segments of the cache do not need to have the same vertical position but might have vertical displacement.

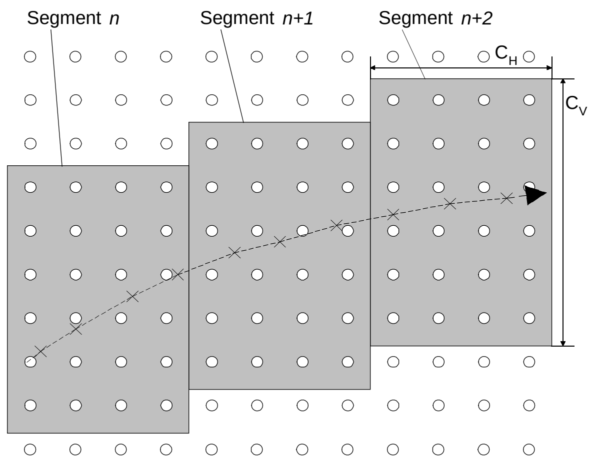

Fig. 8: Cache segments around scanline

The cache segment size is determined by the organization of the external memory and the available bandwidth. As an example, we assume one external DDR-DRAM with a 32bit bus, 200 MHz bus frequency and a burst-length of 8. Then, one DRAM access contains 8*32=256bit, corresponding to ten pixel with 24bit resolution. Thus, a segment width C_H of 10 pixel is sensible.

With a bandwidth of 1280 Gbit/s (32bit * 200MHz * 2 words/clock) and assuming 25% overhead due to task switches, blanking and refresh, an XGA image with 60 Hz frame rate can be transferred eight times. One transfer is needed for writing, so the vertical segment size C_V can be seven.

With the size of the cache segments and the size of the anti-aliasing filter the maximum slope that can be corrected can be determined. Future DDR-DRAMs will have higher bandwidth, thus allowing larger image sizes to be processed and larger vertical segment sizes.

3.5. Brightness Correction

The correction of brightness variation is performed after geometry correction. For each output pixel, a translucency can be defined. The brightness of the output pixel is then reduced by the defined value.

Brightness variations due to the construction of a projection system can be calculated or measured and used for all devices of the production series. Variations due to tolerances or variations of the optical elements can be measured after production.

When geometry distortion is the cause of a brightness variation, the translucency map can be calculated as discussed above. Also, a projector specific translucency map can be modified according to the brightness variation by geometry distortion.

3.6. System Architecture

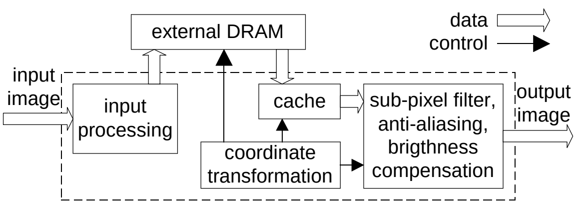

Fig. 9: System architecture

The complete system for compensation of the different optical distortions is depicted in Figure 9.

An example for an image processed with the described architecture is depicted in Figure 10. It shows a combination of keystone and pincushion correction.

4. ASIC Implementation

An ASIC implementation of the presented architecture is currently under design. An FPGA prototype is used for system verification and optimizations.

The design is based on the display controller LEHK–2 [9], which has been successfully implemented as an ASIC and already offers one-dimensional keystone correction. In addition to signal processing features the upcoming ASIC will also contain an embedded CPU and an USB controller to allow the design of a compact and cost-effective display system.

Fig. 10: Processed image

5. References

[1] R. Mohr, B. Trigs, “Projective Geometry for Image Analysis,” ISPRS96, Tutorial, 1996

[2] K. Grüger, G. Bauer, “Datenvisualisierung als 360° Doppelprojektion auf eine Kugel-Innenfläche,” ITG/FKTG-Fachtagung “Multimedia”, 2001 (accepted for publication).

[3] “Daten-/Videoprojektoren - Marktübersicht,” AV-Views 5/2000, pp. 30-37.

[4] “Attraktiv und preiswert - Tests in unserem Labor,” AV-Views 4/2000, pp. 27-31.

[5] G. Wolberg, "Digital Image Warping," IEEE Computer Society Press, 1990.

[6] N. Fliege, "Multiraten-Signalverarbeitung: Theorie und Anwendungen," Teubner, Stuttgart, 1993.

[7] http://www.micron.com

[8] H. Volkers, H. Jeschke, T. Wehberg, "Cache Memory Design For The Data Transport To Array Processors," Proc. of IEEE Int. Symposium on Circuits and Systems, January 1990, pp. 49-52.

[9] M. Winzker, U. Rabeler, M. Stephan, G. v. Husen, "Integrated Display Architecture for Ultra-Portable Multimedia Projectors," SID International Symposium, S. 562-565, 2000.