QR Codes on Curved Media Facades

Two Approaches for Inverse Distortion based on Raytracing and Image Warping

Morin Ostkamp, Sven Luzar and Gernot Bauer

Software Engineering Lab, Münster University of Applied Sciences, Soester Straße 13, 48155 Münster, Germany

Keywords: QR Code, Media Facade, Curved Surface, Raytracing, Image Warping.

Abstract: Media facades and other public displays are a common sight in our everyday life. They can be found at various places, such as shopping malls or traffic hubs, e.g., the Times Square in New York City or Shibuya Crossing in Tokyo. Thanks to advances in technology, media facades are not bound to plane, rectangular shapes. Thus, many media facades are curved to increase their visual appearance or to better harmonize with the building behind. One example of such a curved media facade is the "Medienfassade" in the city of Münster. The shown content, however, is visually distorted due to the facade's concave shape. That visual distortion may cause problems for certain types of content, i.e., QR codes, that may thus not be scanned with mobile apps. Since QR codes became popular in recent years, it appears desirable to display these codes on curved media facades, too. Thus, we propose two preprocessing approaches to compensate for the visual distortions of the curved media facade. An in-situ evaluation showed that both approaches can be used to "rectify" the distorted QR codes and let them be scanned successfully.

1 INTRODUCTION

Nowadays, displays are not restricted to plane, rectangular shapes anymore. Thanks to recent technological advances, e.g., OLED, almost all possible forms and sizes can be realized. On the one hand, this may be an advantage for product designers, as they are less bound to technical restraints. On the other hand, however, this gain in flexibility may cause problems for content designers, as their displayed content may be visually distorted. These visual distortions may not cause problems for most content types. Certain types of content, however, may be affected negatively, i.e., QR codes. Though the specifications of QR codes allow to print them on curved surfaces, most non-industrial scanners, especially smart phones, struggle to scan them. The QR code in Figure 4 (d), for example, cannot be scanned by "ZBar" on an iPhone 4. An example for a curved display is the "Medienfassade" in the city of Münster, which is shown in Figure 1. This media facade has a size of 13.20 m (width) by 13.90 m (height). It is embedded in the front side of a building, 8.70 m above the ground. To better harmonize with the building's architecture and due to restrictions of the city's government, the media facade's surface is concave and set back into the building's body. This causes a noticeable visual distortion, that can be observed in Figure 1. The actual display surface is made of a semi-transparent grid of LEDs with a resolution of 192 x 212 pixels. As QR codes became a popular marketing means on flyers, posters, and billboards in recent years, there was a rising need to show QR codes on the Medienfassade as well. Due to the curved surface, however, usual QR codes, cf. Figure 4 (a), could not be scanned with most smart phone apps, cf. Figure 4 (d). Since there are many more examples of curved media facades and other types of public displays all over the world, e.g., the Times Square in New York City or Shibuya Crossing in Tokyo, we believe that the compensation for visual distortions of QR codes is a significant and important contribution to the community. Thus, we analyzed how to counter the visual distortions of the Medienfassade by following two approaches: (i) raytracing (in the sense of, mathematically speaking, projective geometry) and (ii) image warping. The remainder of this paper is structured as follows: First, we position our research within related work. Then, we present two approaches to let QR codes be scanned successfully on curved surfaces. Next, we present results of an in-situ evaluation of both approaches. We discuss the findings in the following section and conclude the paper with a summary of our contributions and an outlook on subsequent work.

Ostkamp M., Luzar S. and Bauer G..
QR Codes on Curved Media Facades - Two Approaches for Inverse Distortion based on Raytracing and Image Warping.
DOI: 10.5220/0004655904240429
In *Proceedings of the 9th International Conference on Computer Graphics Theory and Applications* (GRAPP-2014), pages 424-429
ISBN: 978-989-758-002-4

Figure 1: The Medienfassade showing a rectangular grid. Due to the facade's curved surface, the grid appears visually distorted, i.e., horizontal lines are bent.

2 RELATED WORK

The work presented in this paper touches upon several research areas. First, we provide an overview of the most important characteristics of QR codes. Then, we present two particular approaches to image processing: The first one is based on raytracing and thus uses a 3D model of the scene shown in the image. The second approach uses 2D image warping techniques instead.

2.1 QR Codes

During the last years, QR codes became popular due to their robust design, high data capacity, and adaptability. A common use case is to print QR codes featuring URLs on flyers, posters, or billboards. Denso Wave developed the two-dimensional bar codes in 1994. The specifications are available as an ISO norm (Denso, 2000) and also in parts online (Denso, 2013). QR codes consist of black and white areas, so called modules. There are different "versions", according to the the number of modules, i.e., the size of the QR code. Figure 4 (a) shows a version 1 QR code containing the word "Test", before applying any transformation described in Section 3. There needs to be an empty area, called "quiet zone", around the code. The size of the quiet zone should be four times as big as a module. If the quiet zone is too small, scanners may fail to scan the code. QR codes can be printed with any color on any background, as long as the contrast between the foreground and the background is kept high. QR codes support four error correction levels: low, medium (used in this paper), quartile, and high. The first one allows 7% of the code to be damaged, while the latter allows to scan codes that are more than 30% damaged. Therefore the contained data is spread redundantly within the code. Furthermore, QR codes are robust against certain (linear or affine) transformations, e.g., rotation or skew. Thus, users can scan codes printed on billboards from almost any position, for example. Though other (non-linear) distortions are also covered by the specifications (Denso, 2013), they may still cause problems, at least for smart phone scanners. Codes printed on coffee mugs, for example, can usually not be scanned with a mobile app. In order to restore the support for smart phones, QR codes can be preprocessed as proposed by this paper.

2.2 Raytracing

Raytracing is a well-established concept in computer vision and graphics (Glassner, 1989). Back in 1968, Appel sparked the development of the raytracing theory in his publication on "Some techniques for shading machine renderings of solids" (Appel, 1968). The basic idea is to trace the path of a ray of light in a virtual scene. Whenever the ray hits an object, the physical effects that can be observed in reality, e.g., refraction, are simulated on a two-dimensional image plane. In subsequent years, the presented algorithms were further evolved and became widely used to generate scene renderings in computer games, for example. Though there may be different raytracing manifestations or implementations, they are all based on mathematical 3D models of the scene that is to be rendered. The creation of such models can be difficult and time-consuming because the reality's details have to be re-created in the virtual space. In this paper, however, the architectural blue prints of the Medienfassade were available and could thus be used to create a suitable 3D raytracing model.

2.3 Image Warping

Image warping is an image manipulation technique that maps points of a source image to specific points in a destination image. One particular application is image morphing, which has been used as a special effect in movies, for example, since it can be used to create smooth transitions between different shapes of a character or an object. Besides that, image warping is also widely discussed in scientific scenarios. Wohlberg (Wolberg, 1996) and Lin and Huang (Lin and Huang, 1999), for example, propose face or facial expression morphing as a means for authentication, speech animation, or natural human computer interfaces. Zhang and Tan (Zhang and Tan, 2002), Liang et al. (Liang et al., 2005), and Meng et al. (Meng et al., 2012) present different approaches for straightening warped text lines in scanned documents. This may help to increase the overall readability for humans as well as the accuracy of machine based optical character recognition (OCR). The undistorted projection of images onto a curved display is also a prevalent use case. Chuang et al. present an approach based on an approximation scheme that can be realized as a hardware component with reasonable performance (Chuang et al., 2010). In contrast to the approach presented in this paper, their system requires an exhaustive specification of environmental parameters, such as the dimensions of the object, which the image is projected on. Their approach is comparable to the one presented in Section 3.1 in this paper. Harville et al. (Harville et al., 2006) and Raskar et al. (Raskar et al., 2005) present self-configuring projector systems that can be used to build large displays on non-planar surfaces. Both systems use cameras to automatically calibrate the projected images and apply image warping in such a way that the final image appears with a minimum local distortion for the viewer. Except for the suggested application scenario, both approaches are comparable to the one presented in Section 3.2 in this paper.

As the review of previous publications reveals, the theoretical basis for the work in this paper has been laid out by the related work cited above. The present paper applies these findings to a specific use case or application scenario and evaluates the feasibility as well as the results.

$$(x-l)^2+(y-m)^2=r^2 \tag{1}$$

$$\begin{aligned} x &= x_1+(x_2-x_1)*t = x_1+i*t \\ y &= y_1+(y_2-y_1)*t = y_1+j*t \\ z &= z_1+(z_2-z_1)*t = z_1+k*t \end{aligned} \tag{2}$$

$$(x_1+i*t-l)^2+(y_1+j*t-m)^2=r^2 \tag{3}$$

$$\begin{aligned} & t^2(i^2+j^2)+t(2ix_1-2li+2jy_1-2mj) \\ & +(x_1^2-2lx_1+l^2+y_1^2-2my_1+m^2-r^2)=0 \end{aligned} \tag{4}$$

$$\begin{aligned} p &= (2ix_1-2li+2jy_1-2mj)/(i^2+j^2) \\ q &= (x_1^2-2lx_1+l^2+y_1^2-2my_1+m^2-r^2)/(i^2+j^2) \\ t_{1,2} &= -p/2 \pm \sqrt{p/2-q} \end{aligned} \tag{5}$$

3 APPROACH

We premised our research on two fundamentally separate approaches: (i) raytracing and (ii) image warping, as described in more detail below. The raytracing approach is based on a theoretical 3D model and may thus not take all real influences, such as camera lens distortions, into consideration. The image warping approach, however, is based on empirical data. We were curious about the outcomes of a comparison of these two opposed approaches.

3.1 Raytracing

Since the Medienfassade has been embedded in a building constructed in 2008, the architectural blue prints were still available. This information could thus be used to create a corresponding 3D model. As a result, the Medienfassade was modeled as a cuboid of 13.20 m width and 13.90 m height, 8.70 m above ground. The cuboid is intersected by a cylinder with a diameter of 21.57 m and an infinite height. The distance between both centers of the cuboid and the cylinder is 22.37 m. The estimated position of the viewer's eye is in 34.00 m in front of the facade and in a height of 1.80 m (supposed the average viewer is about that tall). In the remainder of this paper we are going to refer to this position as the “sweet spot”, since the viewing experience will be optimal from there. Further, the presented approach assumes that the source image, that is undistorted and has not been preprocessed, is held parallel in 0.01 m distance to the viewer's eye. The source image's resolution is twice as high as the resolution of the target, i.e., the media facade, to avoid interpolation issues, that may appear as black spots in the preprocessed image.

The presented approach is based on a minimalistic raytracing implementation. The scene has been modeled as a linked list of the following objects: the world's origin, the viewer's eye (the camera), the source image, the cylinder, and the media facade. The further implementation is based on the following notion. Let the facade's cylinder be defined as in (1), with l and m as the center and r as the radius. A ray, that originates in the viewer's eye, intersects the source image, and finally hits the media facade is defined as in (2), with x_1, y_1, z_1 as its origin and x_2, y_2, z_2 as a second point on the ray specifying the ray's direction. Now, plug (1) in (2) which in turn provides (3). Solving this equation for t provides two solutions. One of them can be ignored ($t < 0$), which provides the result in (4) and (5) (p and q are parts of the p-q-formula to solve the quadratic equation). Plugging (5) back into (2) provides x, y and z coordinates for each

Figure 2: The curved media facade showing a QR Code that has been preprocessed using the raytracing approach.

Figure 3: The curved media facade showing a QR Code that has been preprocessed using image warping.

pixel of the source image on the media facade. Figure 4 (b) shows what the preprocessed image looks like, while Figure 2 displays the visual experience from the sweet spot.

3.2 Image Warping

In order to empirically reproduce the distortion caused by the Medienfassade, we displayed a regular grid on the media facade, see Figure 1. The distorted image of the grid was photographed from the sweet spot. This photograph was used as a calibration image. After cutting and resizing the calibration image in a standard image processing software, it had the same dimensions as the undistorted regular grid. That allowed us to identify the 24 intersection points $\vec{u}_i$ (i=1,...,24) of the grid's lines with their correspondences $\vec{v}_i$ in the calibration image. We then used the correspondence between $\vec{u}_i$ and $\vec{v}_i$ to find a transformation T, that matches any point in the undistorted original image to the corresponding point in the distorted image on the media facade, cf. Equation (6). Note that T has also to fulfill the 24 boundary conditions.

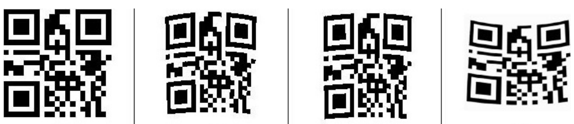
$$\vec{v}_i = T(\vec{u}_i). \tag{6}$$

For our purposes, it is sufficient to assume that the $\vec{v}_i$ in (6) depend on the $\vec{u}_i$ up to third order in the components of the $\vec{u}_i$. In order to find a suitable T we thus have to adjust a set $\beta = (\beta_1, \ldots, \beta_{20})$ of 20 fitting parameters. Equation (6) also takes the form (7), with a remark that $\vec{u}_i = (u_{x,i}, u_{y,i})$. This matrix equation, that is linear in β, can be solved easily, providing the desired transformation T (Whitaker, 2009). Note that the transformation T, that represents the distortion applied to an image when displayed on the media facade, is invertible. That is, if the media facade displays the inversly distorted image $T^{-1}(QR)$ of an original QR code QR, the viewers – and any QR code scanner – will perceive $T(T^{-1}(QR)) = QR$, i.e., the original undistorted QR code, as desired. Figure 4 (c) shows what the preprocessed image looks like, while Figure 3 displays the visual experience from the sweet spot.

3.3 Implementation

To allow for a comprehensive and fast evaluation of the two approaches presented in Sections 3.1 and 3.2, we designed a web front end to generate the individual QR codes. This web front end allows users to enter any string, e.g., texts or numbers, that the preprocessed QR code shall contain. Furthermore, the desired preprocessing method, foreground and background color, as well as the error correction level can be selected. As soon as the user changes any parameter, e.g., the text, the preprocessing is triggered. The result is directly shown in the web front end and can be saved to a local file as well. Since this work is supposed to illustrate the general feasibility of showing QR codes on curved media facades, the implementation is done in a simple, not necessarily most efficient way: The raytracing approach is implemented in PHP, as this programming language integrates well with the web front end. The image warping, however, is done in Java, since there is a well-tested implementation of polynomial image warping available in the "WarpPolynomial" class of the "javax.media.jai" package.

The only apparent difference between both approaches is the time it takes to preprocess an image.



(a) (b) (c) (d)

Figure 4: QR codes samples. (a) Original; (b) Raytracing; (c) Image Warping; (d) Simulated perception of an original QR code on the Medienfassade that cannot be scanned.

$$
\begin{pmatrix}
1 & u_{x,1} & u_{y,1} & u_{x,1}^2 & u_{x,1}u_{y,1} & u_{y,1}^2 & u_{x,1}^3 & u_{x,1}^2u_{y,1}^1 & u_{x,1}^1u_{y,1}^2 & u_{y,1}^3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & u_{x,2} & u_{y,2} & u_{x,2}^2 & u_{x,2}u_{y,2} & u_{y,2}^2 & u_{x,2}^3 & u_{x,2}^2u_{y,2}^1 & u_{x,2}^1u_{y,2}^2 & u_{y,2}^3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
1 & u_{x,24} & u_{y,24} & u_{x,24}^2 & u_{x,24}u_{y,24} & u_{y,24}^2 & u_{x,24}^3 & u_{x,24}^2u_{y,24}^1 & u_{x,24}^1u_{y,24}^2 & u_{y,24}^3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & u_{x,1} & u_{y,1} & u_{x,1}^2 & u_{x,1}u_{y,1} & u_{y,1}^2 & u_{x,1}^3 & u_{x,1}^2u_{y,1}^1 & u_{x,1}^1u_{y,1}^2 & u_{y,1}^3 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & u_{x,2} & u_{y,2} & u_{x,2}^2 & u_{x,2}u_{y,2} & u_{y,2}^2 & u_{x,2}^3 & u_{x,2}^2u_{y,2}^1 & u_{x,2}^1u_{y,2}^2 & u_{y,2}^3 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & u_{x,24} & u_{y,24} & u_{x,24}^2 & u_{x,24}u_{y,24} & u_{y,24}^2 & u_{x,24}^3 & u_{x,24}^2u_{y,24}^1 & u_{x,24}^1u_{y,24}^2 & u_{y,24}^3
\end{pmatrix}
\begin{pmatrix} \beta_1 \\ \vdots \\ \beta_{10} \\ \beta_{11} \\ \vdots \\ \beta_{20} \end{pmatrix}
=
\begin{pmatrix} v_{x,1} \\ v_{x,2} \\ \vdots \\ v_{x,24} \\ v_{y,1} \\ v_{y,2} \\ \vdots \\ v_{y,24} \end{pmatrix}
\tag{7}
$$

The PHP raytracing takes up to 8 seconds on a recent web server, while the image warping only takes 2 seconds on the same machine. This difference, however, is probably due to the varying performance of the employed programming languages and the naive implementation only. Once the images of the QR codes had been preprocessed, the results were saved as bitmap files. Finally, the video editors, who are in charge of the media facade's content, used these bitmap files to include the preprocessed QR codes in the daily feature show.

4 EVALUATION

The Medienfassade is located in a spot of Münster that is highly frequented by passersby, cars, and trains. Thus, preliminary tests prior to public exposure are highly advisable. To verify the preprocessed QR codes in a first step, we used a 3D Blender (`www.blender.org`) model that is based on the same architectural data as presented in Section 3.1. Once the simulation's results seemed reasonable, the same QR codes were tested in-situ. In total, there were three test sessions: (i) a first test with the preprocessed QR codes validated by the Blender simulation; (ii) a subsequent test with refined QR codes based on the findings of the first test; (iii) a final test with optimized QR codes in the afternoon and at night. Different times of the day were chosen since the media facade is exposed to varying lighting due to different weather conditions. The different conditions may have an impact on the contrast and saturation of the shown content, compare Figures 1 and 2. An iPhone 4 and a Samsung Galaxy S3 were used to test whether a QR code could be scanned from the sweet spot.

The third test session showed, that all preprocessed QR codes could be scanned from the sweet spot using either the raytracing or the image warping approach. The width of the sweet spot was about 3.00-4.00 m, while the QR codes had a physical size of approximately 7.50 m on the media facade, cf. Figures 2 and 3. Within the perimeter of the sweet spot, QR codes could be scanned reliably, while most tries failed outside. To verify that the QR codes could be scanned due to the actual preprocessing steps presented above, the system was counterchecked with an ordinary QR code of the same size and content, cf. Figure 4 (a). As expected, the distorted QR code, cf. Figure 4 (d), could not be scanned from any position with none of the testing devices.

The tests also showed that QR codes, which do not use prime colors, i.e., red, green, and blue, for the foreground as well as the background color, cannot be scanned. In contrast to this large impact of the chosen colors, the lighting conditions, i.e., day vs. night, as well as defective LEDs did not have a noticeable influence.

5 DISCUSSION

As the in-situ evaluation revealed, both presented preprocessing approaches – the theoretical as well as the empirical – are viable means to display QR codes on curved media facades so that they can be scanned with smart phones.

Both approaches, however, are limited to a predefined sweet spot. Once the viewer leaves this sweet spot, the visual distortions of the curved surface are not completely compensated anymore so that the QR code cannot be scanned. This shortcoming, however, could be mitigated using the raytracing approach: Preprocessed QR codes could be generated on the fly for any sweet spot, thanks to the detailed 3D model, if the implementation's performance could be improved. Even then, however, the shown QR code could still be scanned from only one sweet spot at a time. To alleviate this, cameras could be used to locate the largest group of viewers, causing the system to generate QR codes for that particular viewing position.

Though the raytracing's flexibility is advantageous on the one side, the need for a 3D model renders this approach less general and more complex on the other side. To remedy this, depth scanners etc.

could be used to derive sufficient models more easily. However, defining the correspondences for the image warping approach can be done easily using standard image processing software and may thus be the more feasible approach for real scenarios.

With regards to the relatively low resolution of the media facade at hand, the mandatory "quiet zone" takes up a significant amount of display real estate. In turn, that display area becomes unavailable, though it could be used more efficiently, e.g., to display a larger QR code. This loss of display space could be countered by reducing the error correction level to a minimum. However, this is only a feasible step, if the overall quality of the shown QR code is reasonably high, e.g., there are no defective pixels. Which version of a QR code is available for specific media facades in general depends on the facade's resolution and other characteristics, such as the physical size of a pixel.

Though the specifications allow QR codes to be printed in many color combinations (as long as the contrast between the foreground and the background is high enough), there appears to be a constraint when displaying QR codes on an LED media facade. Apparently, the media facade blends colors using pulse width modulation (PWM) if the color is not a "true" red, green, or blue. To display a 75% red, for example, the media facade turns on the red LEDs for 75% of a cycle and turns them off for 25% of the time. These fast changes are not perceptible for the human eye, but become visible in a picture taken by a camera. These fluctuations apparently influence the scanning process of QR codes heavily.

6 CONCLUSION

In this paper, we presented two approaches to preprocess images of QR codes in such a way that the codes can be scanned successfully on the surface of a curved media facade. Though there are some limitations to both approaches, they proved to work well in the given application scenario.

The contribution of this paper is relevant since QR codes become increasingly important in numerous use cases, e.g., pedestrian indoor and outdoor navigation, advertising, or mobile internet access – many more lie undiscovered. At the same, the number of media facades and other types of digital signage is rising (Schaeffler, 2008).

To further elaborate this research area, we would like to focus on the following aspects: The raytracing performance could be improved by porting the approach to a more powerful programming language, e.g., C or C++, and make use of dedicated hardware, such as the GPU on recent graphics cards. Moreover, it would be desirable to increase the perimeter of the sweet spot or to avoid such a designated viewing location in the first place. We would also like to further investigate the maximal and minimal sizes of QR codes with regards to the media facade's resolution and the users' viewing position.

REFERENCES

Appel, A. (1968). Some techniques for shading machine renderings of solids. In *Proc. AFIPS '68*, pages 37–45. ACM.

Chuang, Y.-M., Hsu, S.-P., Huang, K.-C., Chang, Y.-C., and Ruan, S.-H. (2010). Implementation of image warping with application to projection onto a curved display. In *Proc. ICIEA '10*, pages 404–408.

Denso (2000). Information technology – automatic identification and data capture techniques – qr code 2005 bar code symbology specification. Technical Report ISO/IEC 18004:2006, International Organization for Standardization.

Denso (2013). Qr code standardization. www.denso-adc.com/dl/1, accessed: 31.10.2013.

Glassner, A. S., editor (1989). *An introduction to ray tracing*. Academic Press Ltd., London, UK.

Harville, M., Culbertson, B., Sobel, I., Gelb, D., Fitzhugh, A., and Tanguay, D. (2006). Practical methods for geometric and photometric correction of tiled projector. In *Proc. CVPRW '06*, pages 5–5.

Liang, J., DeMenthon, D., and Doermann, D. (2005). Flattening curved documents in images. In *Proc. CVPR '05*, volume 2, pages 338–345.

Lin, D.-T. and Huang, H. (1999). Facial expression morphing and animation with local warping methods. In *Proc. ICIAP '99*, pages 594–599.

Meng, G., Pan, C., Xiang, S., Duan, J., and Zheng, N. (2012). Metric rectification of curved document images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4):707–722.

Raskar, R., van Baar, J., Beardsley, P., Willwacher, T., Rao, S., and Forlines, C. (2005). ilamps: geometrically aware and self-configuring projectors. In *SIGGRAPH '05 Courses*. ACM.

Schaeffler, J. (2008). *Digital Signage: Software, Networks, Advertising, and Displays: A Primer for Understanding the Business*. Focal Press, 1 edition.

Whitaker, R. T. (2009). Geometric transformations and image warping. *University of Utah, CS6640, www.coe.utah.edu/~cs6640/geometric_trans.pdf (2013/06/12)*.

Wolberg, G. (1996). Recent advances in image morphing. In *Proc. CGI '96*, pages 64–71.

Zhang, Z. and Tan, C. (2002). Straightening warped text lines using polynomial regression. In *Proc. Image Processing '02*, volume 3, pages 977–980.