Automatic Correction of Lens Distortion by Using Digital Image Processing

Harri Ojanen

July 10, 1999

Introduction

Many inexpensive camera lenses, especially zoom lenses, suffer from very clear distortion. We show that even with such a lens near perfect images can be obtained by using a correction algorithm based on a simple mathematical model and without any special measuring tools. See Figures 9–12 for real-life examples.

The most typical cases of distortion, barrel and pincushion distortion, are illustrated in Figures 1 and 2. In practice the distortion is often more complicated, a combination of both of these cases, Figure 3.

Background

In both the object and film planes we use radial coordinates so that the origin is at the point where the optical axis meets the plane. The distance r in the planes is normalized so that in both planes $r = 1$ corresponds to the corners of the object or the image.

The lens mapping $L(r)$ is used to describe how a lens distorts an image. We define the mapping for a fixed object plane distance as follows: a point in the object plane a distance r from the optical axis is imaged by the lens to a point a distance $L(r)$ from the axis in the film plane. A lens with no distortion would have $L(r) = r$. In general, the mapping may depend on the focus distance; for zoom lenses in practice it always depends on the focal length.

Transverse magnification at a distance r from the optical axis is given by $M = dL/dr$. Barrel distortion results when the magnification decreases with the off-axis distance r, that is, $dM/dr < 0$. Conversely, pincushion distortion is the result of magnification increasing with the off-axis distance, $dM/dr > 0$. It is possible to have both types of distortions in a lens. Figure 3 shows a hypothetical example, see Figures 6, 9, and 7 for a real example. (For a more traditional discussion of distortion see [1].)

We introduce the undistortion function $U^{-1} = L$. It is convenient to write $U(r) = r + \Delta(r)$. If $\Delta(r) = c_2 r^2 + c_3 r^3 + \cdots$, we have $U(r - \Delta(r)) = r - c_2 r^2 + c_2 (r - \Delta(r))^2 + \mathcal{O}(r^3) = r + \mathcal{O}(r^3)$. Thus we get approximate formulae in terms of $\Delta(r)$ for the lens function and magnification: $L(r) \approx r - \Delta(r)$ and $M(r) \approx 1 - \Delta'(r)$.

The Model

For practical reasons the model used in the optimization is not just the undistortion function $U(r) = 1 + \Delta(r)$ but in fact a composition of the following mappings:

- Translation $T_{w_1}(z) = z + w_1$ in the film plane to allow for the possibility that the optical axis of the lens does not pass through the center of the image (e.g., due to incorrect cropping during scanning of a slide).

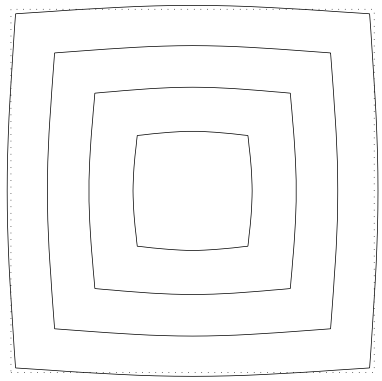

Figure 1: Barrel distortion.

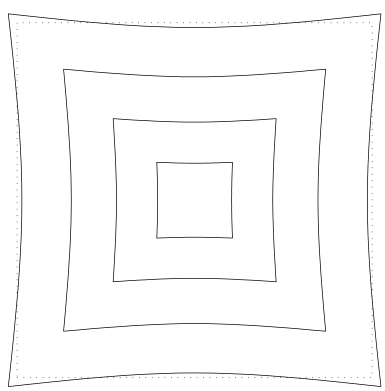

Figure 2: Pincushion distortion.

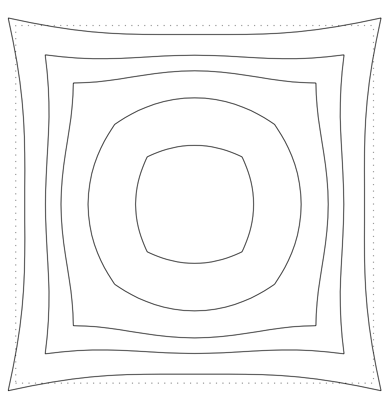

Figure 3: Combination of barrel and pincushion distortions.

- The non-linear undistortion function $U(r) = r + \Delta(r)$ to correct the distortion in the lens. Low order polynomials in r as well as trigonometric polynomials were used for $\Delta(r)$.

- If the film plane was not parallel to the target sheet the image shows perspective. This is compensated for by computing the effect of perspective through a pinhole and varying the parameters of the pinhole: This is accomplished with two parameters, the vectors $\mathbf{n}$ and $\mathbf{P}$. The film plane is perpendicular to the vector $\mathbf{n}$ and passes through the point $\mathbf{P}$, which is also the projection of the pinhole to the film plane. The pinhole is at a distance $|\mathbf{n}|$ from the film plane. (Note that varying $|\mathbf{n}|$ allows changing the overall magnification of the image.) The object plane is taken fixed. Deriving the formulae for the mapping $P_{\mathbf{n},\mathbf{P}}$ from the film plane to the image plane is straightforward but tedious, hence is omitted.

- The last two mappings are rotation of the target sheet $R_\theta(z) = e^{i\theta} z$ and translation of the target sheet $T_{w_2}(z) = z + w_2$.

Hence the composite mapping taking a point in the film plane to the abstract object plane is

$$F(z) = T_{w_2} \circ R_\theta \circ P_{\mathbf{n},\mathbf{P}} \circ U \circ T_{w_1}. \qquad (1)$$

Note that only $U(r)$ is relevant for correcting images, the other functions are needed simply to compensate for inaccuracies in the test setup.

Optimization

The test sheet that is photographed consists of a rectangular grid of black dots, see Figure 5. An example of a photograph of the sheet is shown in Figure 6. From a digital image it is easy to compute the locations of the centers of the dots in the distorted image, we call these $\{z_i\}_i$. The goal is then to choose the parameters of the mapping F in (1) so that F maps the set $\{z_i\}_i$ to a rectangular grid of

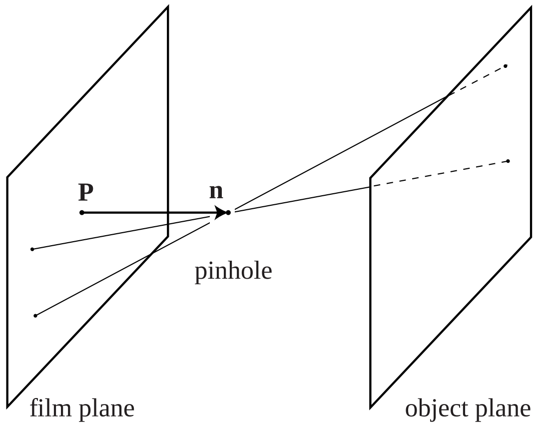

Figure 4: Mapping rays through a pinhole.

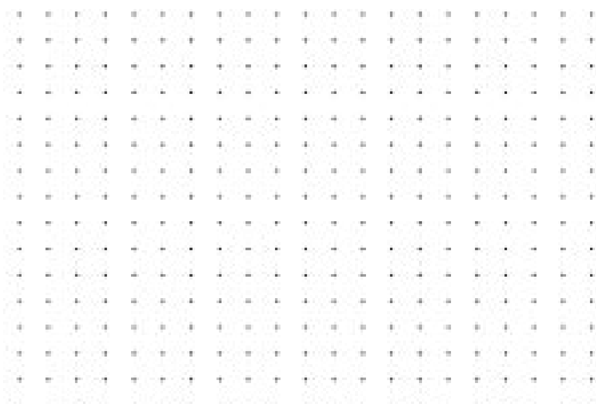

Figure 5: Target sheet

points $\{Z_i\}_i$. This is done in the least-squares sence, that is, by minimizing $\sum_i |F(z_i) - Z_i|^2$. The actual computations were done by using the Levenberg-Marquardt algorithm [2].

The interesting output of the algorithm is the parameters for the function $\Delta(r)$.

This approach answers the question "how should the image be modified to make it appear undistorted?". We could as well have worked with the inverse mapping of F (certainly each factor in (1) is invertible). Recall that $U^{-1} = L$, so optimizing $\sum_i |z_i - F^{-1}(Z_i)|^2$ yields direct information about the lens mapping L, that is, answers the question "how does the lens distort the image?". Convenience dictates which direction is more practical.

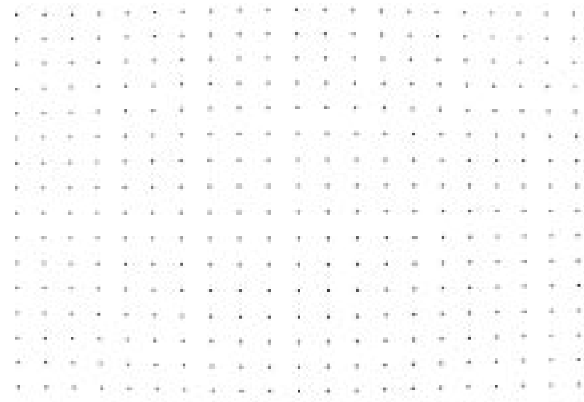

Figure 6: Target sheet photographed with a 20mm lens.

(The first approach described is more direct with the image processing software used.)

Software

The software was written using mostly Matlab. There are three main parts: initialization, optimization, and correction. The initialization part computes the centers of the dots in the distorted image and passes them to the optimization part.

The result of the optimization, essentially the function $\Delta(r)$, contains all the information that is needed to undistort images created with a particular lens. A stand-alone C program was written for correcting images based on the parameters of $\Delta(r)$. Also, the software can create "displacement maps" for Adobe Photoshop, so the correction can be done in any image processing program has a Photoshop compatible displace filter.

To apply corrections to images only the C program or a displacement map is needed.

The software should be accessible to a wide audience: the Matlab part is written so that it works also in the student version of Matlab and a bare-bones version compatible with GNU Octave is also available.

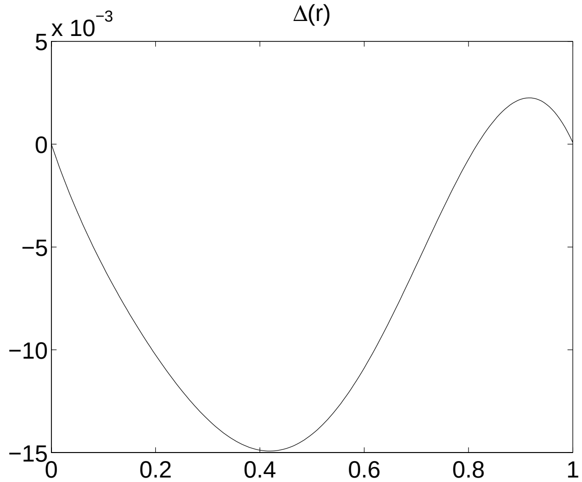

Figure 7: $\Delta(r)$ for the lens used in Figure 7.

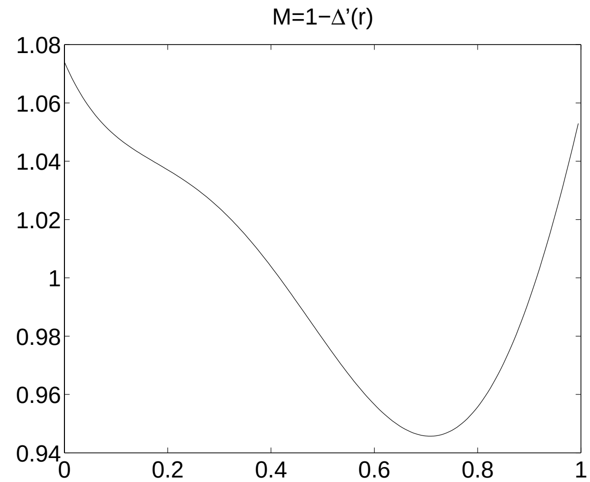

Figure 8: Transverse magnification $M(r)$.

Examples

Figures 9 and 10 show an example of a zoom lens at the 20mm setting both before and after correction. The function $\Delta(r)$ for this lens is shown in Figure 7 and the transverse magnification in 8. Note how dM/dr changes sign, hence the lens has both barrel and pincushion distortion. This is also clear comparing the top of the building in Figure 9 with Figure 3: note the similarity with the middle "square" in Figure 3.

A second example of a different zoom lens at the 28mm setting is shown in Figures 11 and 12.

Conclusions

Lens design is inherently a process of compromise between not only the specifications of the lens but also different aberrations. In many cases it appears that high distortion is tolerated when low cost or long zoom ranges are desired. The method described here shows, fortunately, that distortion is very easy to correct digitally.

Even with a rudimentary setup we have achieved near perfect results. With a complete description of the lens ray tracing techniques could be used to compute $\Delta(r)$ even more accurately.

The undistortion process could be incorporated directly into the firmware of a digital camera. This would allow more freedom in designing the lens but still allow excellent image quality.

References

[1] E. Hecht, *Optics*, 3rd ed., Addison-Wesley, 1998.

[2] D. W. Marquardt, *An algorithm for least-squares estimation of nonlinear parameters*, J. Soc. Indust. Appl. Math. **11** (1963), 431–441.

email: `Harri.Ojanen@iki.fi`
web: `www.iki.fi/Harri.Ojanen/`

Figure 9: Original image formed with a 20mm lens.

Figure 10: Corrected image.

Figure 11: Image formed with a 28mm lens.

Figure 12: Corrected image.