

Imatest - Distortion

New in Imatest 3.4 *Distortion can analyze edges near the image boundary (in addition to lines).*

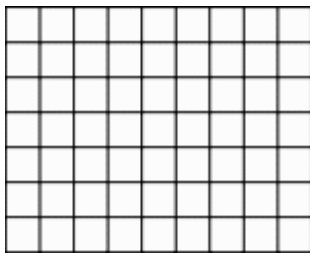
Distortion is also measured in [SFRplus](#): results are in the Image & geometry display. The Distortion module uses the entire pattern to analyze distortion while SFRplus uses only the top and bottom bars. Results are generally close, especially for SMIA TV distortion.

Distortion

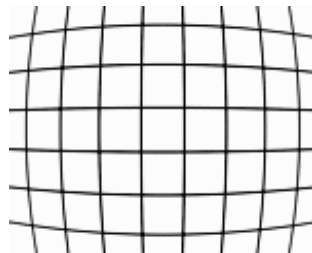
- measures radial lens distortion, an aberration that causes straight lines to curve,
- calculates coefficients for removing it using [nonlinear optimization](#), and
- provides additional information on geometric distortion in digital images.

Distortion calculates the coefficient for correcting distortion using [Picture Window Pro](#), which uses a tangent/arctangent distortion model. A procedure for obtaining approximate [Ptlens](#) coefficients is described [below](#).

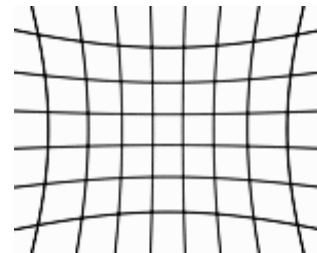
Lens distortion has two forms, **barrel** and **pincushion**, as illustrated below.



None



Barrel



Pincushion

Distortion tends to be most serious in extreme wide angle, telephoto, and zoom lenses. It is most objectionable in architectural photography and photogrammetry — photography used for measurement (metrology). It can be highly visible on tangential lines near the boundaries of the image, but it's not visible on radial lines. In a well-centered lens distortion is symmetrical about the center of the image. But lenses can be [decentered](#) due to poor manufacturing quality or shock damage.

In the simplest lens distortion model, the undistorted and distorted radii r_u and r_d (distances from the image center normalized to the center-to-corner distance (half-diagonal) so that $r = 1$ at the

corner) are related by the equation,

$$r_u = r_d + k_1 r_d^3 \quad \text{where } k_1 > 0 \text{ for barrel distortion and } k_1 < 0 \text{ for pincushion.}$$

This third-order equation is one of the [Seidel Aberrations](#), which are low-order polynomial approximations to lens degradations. Other aberrations include astigmatism, coma, curvature of field, etc. The third order approximation is sufficient for many lenses, but Imatest also calculates the fifth-order coefficients, which can be more accurate for certain lenses, for example, for the [Sigma 18-125mm zoom](#) at 18mm and the [Zeiss 21mm f/2.8 Distagon](#).

$$r_u = r_d + h_1 r_d^3 + h_2 r_d^5$$

The fifth-order equation can quantify “wave” or “moustache” distortion, which might, for example, resemble barrel near the center of the image and pincushion near the corners. [Distortion](#) by Paul van Walree is excellent background reading.

To measure distortion, you’ll need a rectangular or (preferably) a square grid pattern, which you can create using [Test Charts](#). Print the chart, photograph it, and enter the image into the Imatest Distortion module, as described below.

Instructions

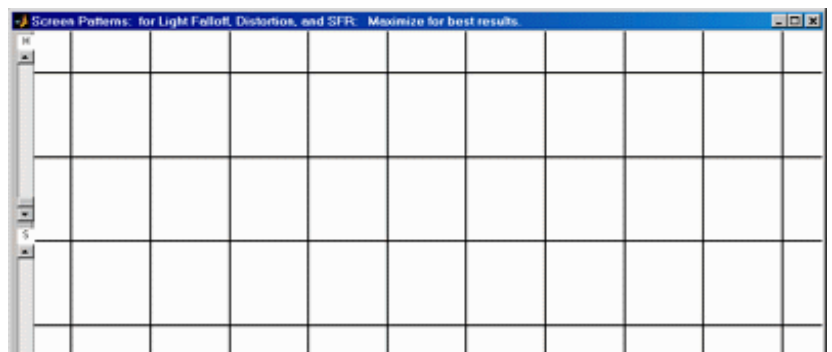
- **Test chart.** You can use any chart with a square or rectangular grid or a square (checkerboard) pattern.

You can create your own using

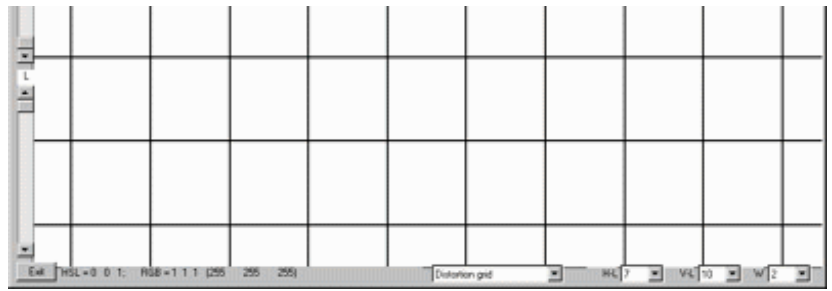
- the Distortion grid or checkerboard pattern the Imatest [Test Charts](#) module, or
- the Imatest [Screen Patterns](#) module (Distortion grid or Squares (checkerboard)).

Screen Pattern module for Distortion.

*The **Screen Patterns** display, shown on the right, works best with large flat screen LCDs (it’s **not** recommended with CRTs).*



Click on on the right the Imatest main window, select Distortion grid or Squares (checkerboard) in the box at the bottom-center, then maximize the window. You can select the number of Horizontal and Vertical lines as well as the line width (in pixels). For best results the pattern should be square.



Test Charts dialog box for preparing a file to print the distortion grid.

- A portion of the [Test Charts](#) input dialog box is shown on the right, with recommended settings:

- **Pixels per inch (PPI).** 100-300. High resolution isn't necessary because Distortion doesn't measure sharpness.
- **Type** (grid or checkerboard) selects chart type.
- **Horizontal divisions.** 10-15 should work for most situations. For extreme distortion, you will either need to use fewer divisions or select an ROI (i.e., crop the image).
- **Vertical divisions.** Square is recommended in most cases.
- **Line width (pxls).** (Note: with the recommended checkerboard chart, line width is not an issue.) Only applies when Type is set to Grid. Typically in the range of $.01 \times \text{PPI}$ to $.025 \times \text{PPI}$. The printed line should be clear (not pale), but not too thick. The relative line widths in the preview (to the right of the selection boxes) is not to scale. They are (relatively) thicker than in the final image.

The line width should be thick enough to cover at least two pixels in the captured image.

Narrower lines may result in errors (Inconsistent number...). For example, suppose your captured image were 640x480 pixels (VGA). The line width (as shown on the right) would be 3 pixels out of a chart width of 1476. In the captured chart, the line width would be $640 \cdot (3/1476) = 1.3$, **inadequate for this small image size**, though this chart would be perfectly adequate for 1+ megapixel cameras.

- **Gamma**, and Contrast are turned off for the Grid pattern.
- **Print the test chart file.** Size isn't critical, but larger is better. (13x19 inches or larger is recommended if your printer can handle it). Matte paper is fine. Handmade charts are acceptable if they're **carefully** done. The print should be made from an image editor using the saved TIFF file, **not** the Matlab preview.
- **Photograph The chart.**
 - It should be evenly lit (though lighting is less critical than with [Light Falloff](#)). Lighting instructions are found in [Using SFR, Part 1](#).

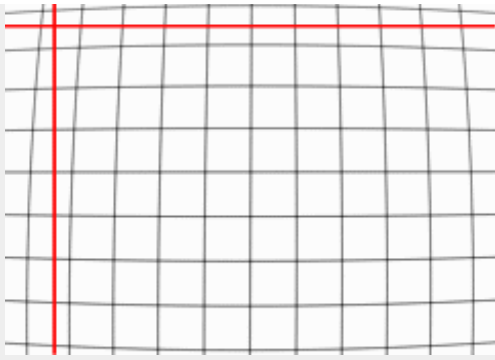
Be especially careful to avoid glare "hot spots" on chart images. Unlike SFR, where small regions can be selected to avoid hot spots, glare hot spots are likely to interrupt grid lines, which will terminate Distortion analysis. Glare can be especially troublesome with wide angle lenses. To minimize glare we recommend printing Distortion charts on matte surfaces.

- The chart should be clean. Distortion will terminate if it mistakes spots in the image for lines. If needed, spots can be manually removed using an image editor "clone" function.
- Set exposure compensation to overexpose by one to two f-stops so white areas are light (not middle gray) and line contrast is sufficient.
- The camera should be pointed straight at the chart. (The optical axis of the lens should be normal to the chart surface.) Small amounts of tilt and perspective distortion (convergence of lines; not a lens aberration) are tolerated, but large amounts may cause Distortion to terminate.
- The camera should be set to a low ISO speed to minimize noise. This is especially important for compact digital cameras, where noise can become severe at high ISO speeds.

Good image

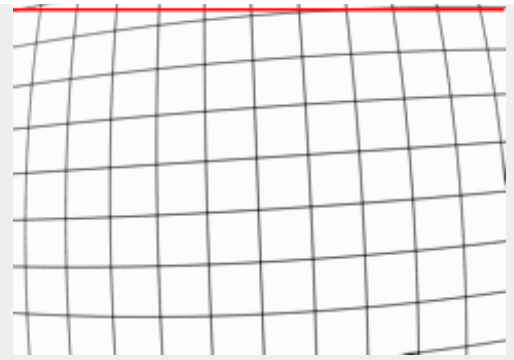
Bad image

Horizontal line



Horizontal and vertical lines should fit between lines in the image for the lines to be detected reliably.

cannot fit between all lines in the image, compromising the reliability of line detection. This image is usable if it is cropped (i.e., a Region of



interest is selected).

Imatest versions 2.3.6+ (released May 2007) performs better with images of this type with ROI filtering set to weak (frequently without cropping), but careful orientation (as shown in the image on the left) is recommended for the most reliable results.

Save the image. High quality JPEG format is preferred because it preserves EXIF data in a format Imatest can read. There is no need for RAW quality.

Resize the image? Because Matlab uses double precision for most math operations, each 24-bit color pixel requires 24 **bytes**. A 6.3 megapixel image can take 151 megabytes of memory (ouch). This can slow down computations in computers that don't have gobs of memory. (It can even bog down computers with 512k of memory.) We are looking into solving this problem. In the meantime we recommend resizing images to about 1000 pixels wide in an image editor. (Any editor will do.) Resizing has no effect on the results.

Launch Imatest. Click on .

Very large files (height x width x colors over 40 MB) may cause memory overflow problems. Files over 40 or 80 MB can be automatically reduced 1/2x linearly (using 1/4 the memory). Click Settings, Options and Settings... (in the Imatest main window) and make the appropriate setting in LARGE FILES (Light Falloff, Distortion).

Open the image file.

Batch mode This module can operate in batch mode in Imatest Master, i.e., it can read multiple input files. All you need to do is select several files using the standard Windows techniques of shift-click or control-click. There are three requirements for the files.

They should (1) be in the same folder, (2) have the same pixel size, and (3) be framed identically.

The input dialogs (cropping (if applicable), settings, and save) are the same for the first run as for standard non-batch runs. Additional runs use the same settings as the first run. Since no user input is required they can run extremely fast.

If the order of the files in a batch runs is different from the selection order, click Settings, Options and Settings... (in the Imatest main window) and change the setting in Batch run order.

One caution: Imatest can slow dramatically on most computers when more than about twenty figures are open. For this reason we recommend checking the **Close figures after save** checkbox, and saving the results. This allows a large number of image files to be run in batch mode without danger of bogging down the computer.

Crop the image (select the ROI) if needed using the usual clicking and dragging technique. Cropping may be helpful if horizontal or vertical lines are not entirely within the ROI (region of interest), though Imatest 2.3.6+ often works well in such cases. An example is shown below in the section on [Severe distortion](#). Click outside the the image to select the entire image. You can select either a grid pattern or a single line, which is useful in patterns such as the ISO 12233 chart, which contains two lines suitable for Distortion: one nearly vertical and one nearly horizontal (both slightly tilted). One of them is illustrated below. Make sure to leave some breathing room around the line. In selecting the crop for grid patterns (or deciding not to crop) try to choose a crop where no lines cross the crop boundaries, i.e., are partly inside and partly outside. The horizontal (well... sort of) line at the bottom right of the “**Bad**” image above is caused Distortion to terminate in Imatest versions earlier than 2.3.6, but is very likely to work will with 2.3.6+.

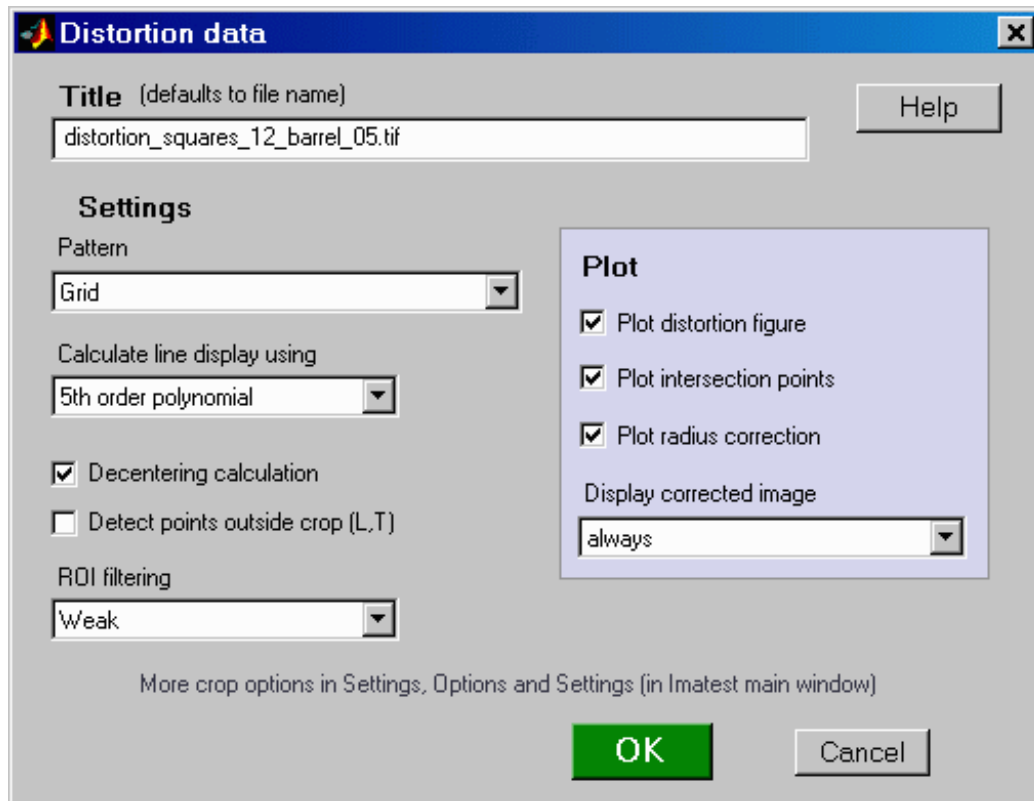
Three cropping (ROI selection) options are available by clicking Settings, Options and Settings... in the Imatest main window. These include

Never crop.
Select crop by dragging cursor. Ask to repeat crop for same sized image.
Select crop by dragging cursor. Do not ask to repeat crop.

Although the first option (Never crop...) was the default in early version of Distortion,

the second option (**Select crop by dragging cursor. Ask to repeat crop for same sized image**), which is similar to the ROI selection in SFR, may be preferred in many cases.

The input dialog box, shown below, appears. Title defaults to the file name; you may change it if needed. The most important choice is the algorithm that detects pattern location. Figures can be selected in the Plot box on the right. **Plot intersection points**, **Plot radius correction**, and **Display corrected image** appear in Imatest Master only.



Distortion input dialog box

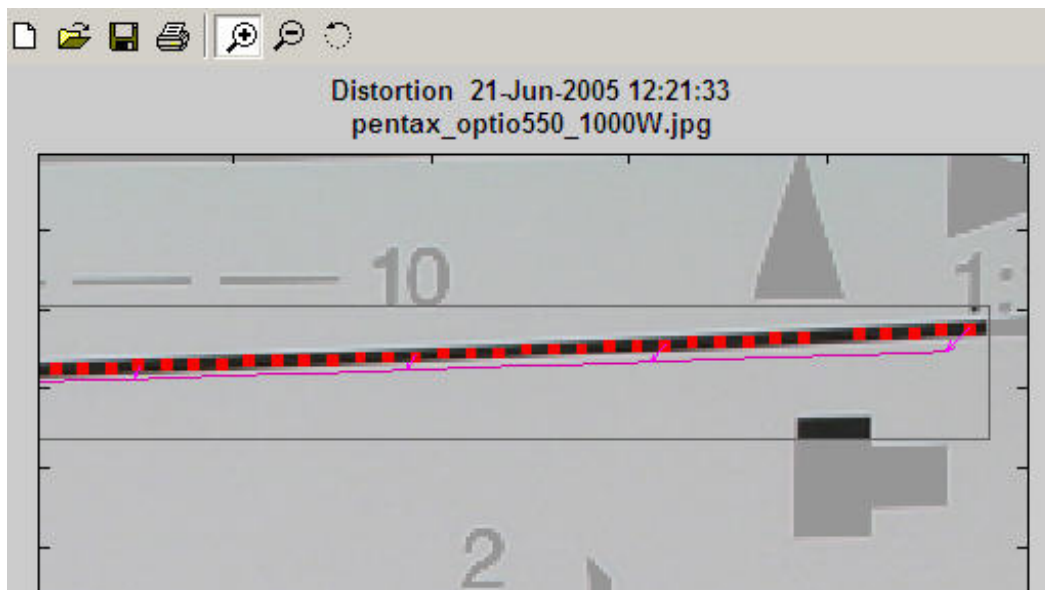
- Pattern
 - **For grid patterns**, choose Grid.
 - **For lines**, choose Single line (some interference). This is the appropriate choice for the usable lines in the ISO-12233 pattern. “Interference” refers to extra detail that may be present near the line (lettering, etc.) This choice will work for lines near the boundaries of a grid pattern. An example from an ISO-12233 chart is shown below.
 - **For square/checkerboard patterns**, choose Squares (checkerboard). This pattern, introduced with Imatest 2.7.1, may be slightly more robust than the grid.



- **Calculate line display using** selects the formula for plotting the corrected lines. Useful for comparing the different algorithms. Choices are 3rd order (coefficient k_1 , [below](#)), 5th order (coefficients h_1 and h_2 , [below](#)), and PW Pro (Picture Window Pro) tan/arctan (coefficient p_1 , [below](#)).
- **Decentering calculation** is described [below](#). It is optional because it takes extra time. It can only be selected for the Grid pattern. If the box is unchecked, distortion is assumed to be symmetrical around the geometric center of the image.
- **ROI filtering** When set to **Strong**, this setting filters out ROIs that might cause Distortion to crash — in Imatest versions earlier than 2.3.6. We recommend setting ROI filtering to Weak since many boundary conditions that formerly crashed Imatest now work well.
- **Plotarea**. Selects which plots to produce.
 - **Plot distortion figure** selects the [main distortion figure](#).
 - The **Plot intersection points** and Detect points outside crop checkboxes are described [below](#). If the first of them is checked, another figure is produced. They only work with the Grid pattern.
 - **Plot radius correction** is described [below](#).
 - **Display corrected image** creates an additional figure with the corrected image calculated with the algorithm selected in Calculate line display using, above. It is most valuable for highly distorted lenses where the input image has been cropped. Choices are never, crop only (the default), and always. An example is shown below in the section on [Severe distortion](#).

Click . The calculations are performed, and the results figure(s) described in [Results](#) appears. You can zoom in by clicking on (and highlighting) the magnifier icon , then clicking on portions of the image of interest, as shown below. The detected points on the image are shown as **red** squares. The straightened line is drawn in **magenta**. Double-clicking restores the entire image.





An enlarged portion of the results figure.

Save results dialog box

When the calculations are complete the Save dialog box appears. The default directory is subdirectory Results in the data file directory. You can change to another existing directory, but new results directories must be created outside of Imatest—using a utility such as Windows Explorer. (This is a limitation of this version of Matlab.) The selections are saved between runs. You can examine the output figures before you check or uncheck the boxes. Select the items you wish to save, then click or . File names (where *filename* is the input file name):

[*filename*]._distortion.png

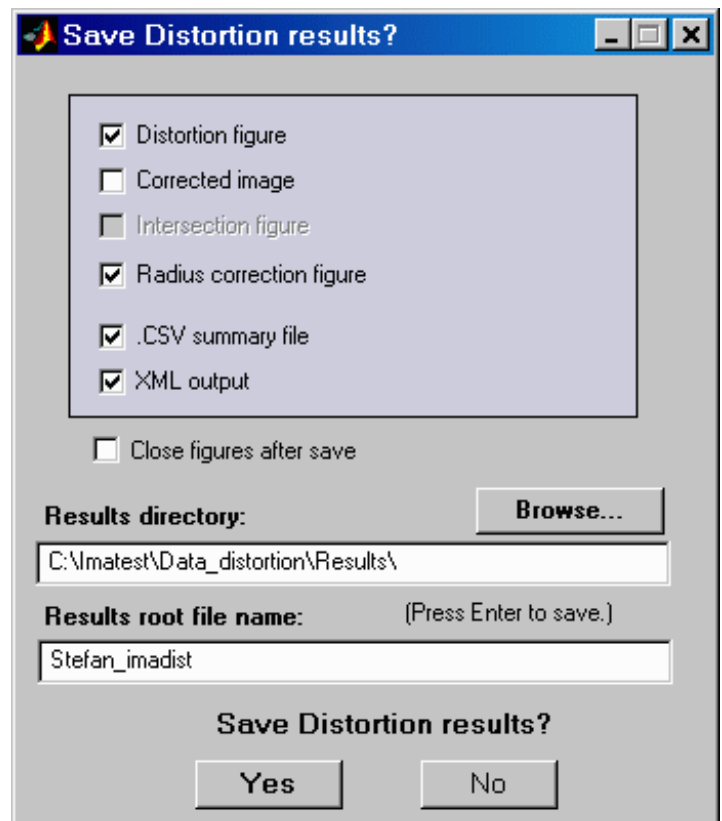
[*filename*]._corrected.png (only if displayed)

[*filename*]._intersections.png

[*filename*]._radius_corr.png

[*filename*]._summary.csv

[*filename*].xml



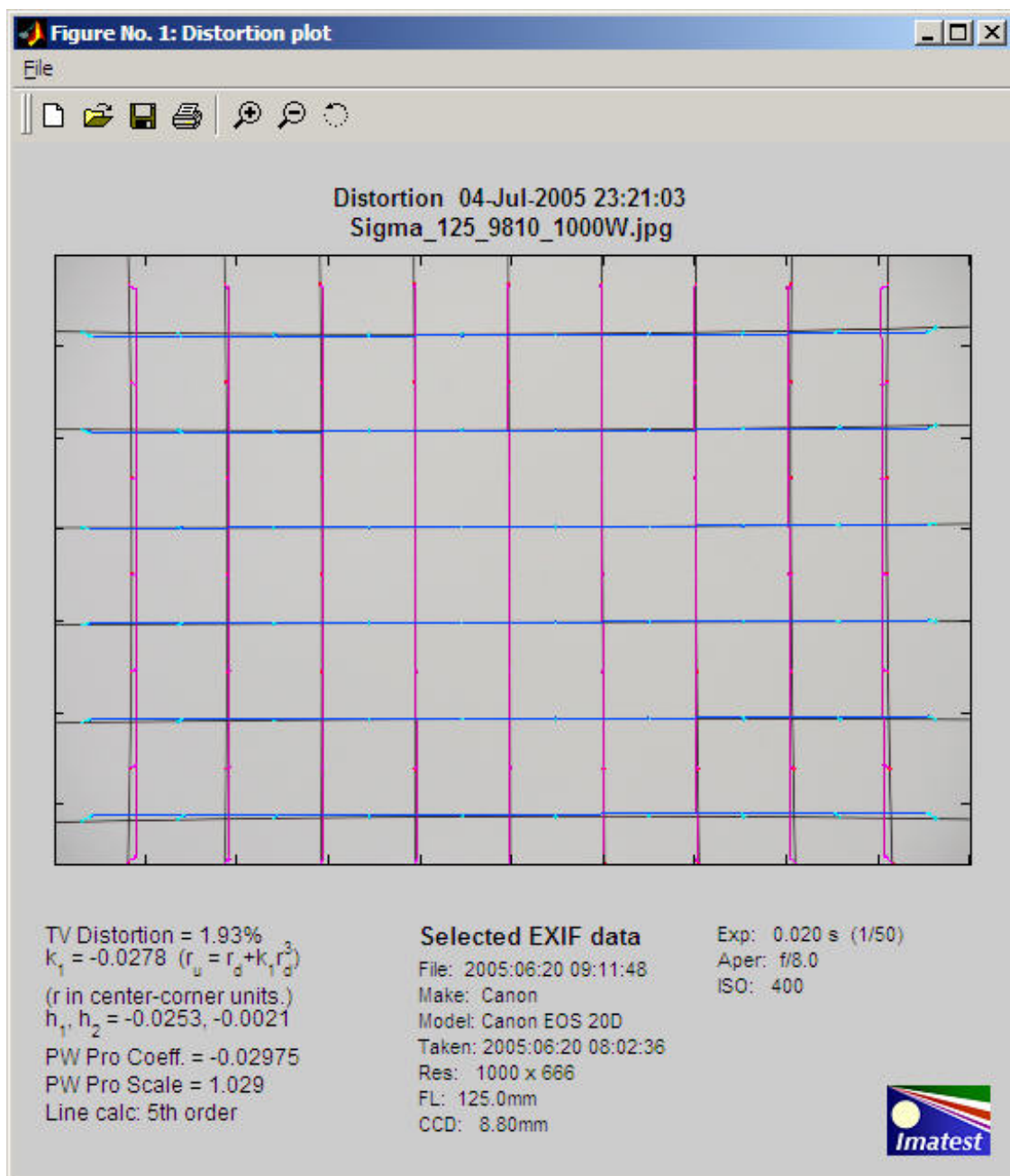
The root file name (*[filename]*, above) defaults to the image file name, but can be changed using the Results root file name box. Be sure to press enter. Checking **Close figures after save** is recommended for preventing a buildup of figures (which slows down most systems) in batch runs.

The CSV and XML files contain **EXIF data**, which is image file metadata that contains important camera, lens, and exposure settings. By default, Imatest uses a small program, jhead.exe, which works only with JPEG files, to read EXIF data. To read detailed EXIF data from all image file formats, we recommend downloading, installing, and selecting [Phil Harvey's ExifTool](#), as described [here](#).

Results

Main figure

An example of Distortion output is shown below for the [Sigma 18-125 mm f/3.5-5.6 DC lens](#) (designed for APS-C-sized sensors, such as the Canon EOS-10D, 20D, Digital Rebel, Nikon D100, D70, etc. The Sigma is an excellent lens — a bargain — except for its autofocus. Mine doesn't autofocus as reliably as Canon lenses, but it works beautifully on manual. The autofocus problem is plainly visible when working with the distortion chart.



Main figure showing modest pincushion distortion

The Sigma has modest amounts of pincushion distortion at 125 mm and barrel distortion at 18mm, its widest angle setting. Corrected vertical lines are **deep magenta**; horizontal lines are **blue**. The following results are displayed on the left below the image.

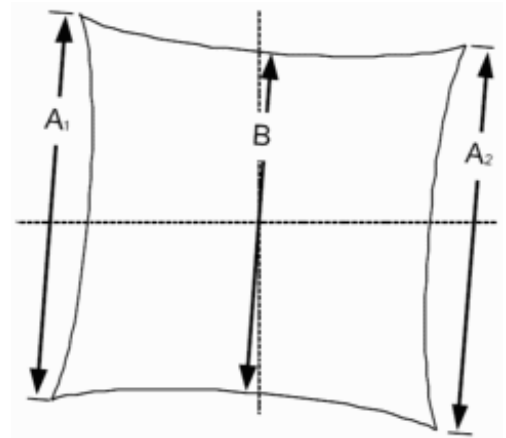
SMIA TV Distortion

- **TV Distortion** from the [SMIA](#) specification, §5.20.

Referring to the image on the right,

$$\text{SMIA TV Distortion} = 100(A - B) / B; \quad A = (A_1 + A_2) / 2$$

The box on the right is described in the SMIA spec as “nearly filling” the image. Since the test chart grid may not do this, Distortion uses a simulated box whose height is 98% that of the image. Note that the sign is opposite of k_1 and p_1 . SMIA TV Distortion > 0 is pincushion; < 0 is barrel.



Algorithm: SMIA TV Distortion is not actually calculated from the upper and lower vertical lines in the chart, whose locations can vary considerably in different images. Instead it is calculated from the [distortion coefficients](#): k_1 prior to Imatest 3.7; h_1 and h_2 (slightly more accurate) for 3.7+, using virtual horizontal lines located 1% of the image height below the top and above the bottom of the image. Note that **all** lines in the image are used to calculate the k and h coefficients. This sometimes results in a discrepancy with the distortion measured from just the top and bottom horizontal lines, particularly in the rare (but not exceptional) case where horizontal and vertical lines have different amounts of distortion.

SMIA vs. traditional TV distortion

Thanks to Mark Butterfield for pointing out that **SMIA TV distortion is twice as large (2X) as traditional TV distortion**. The traditional definition, shown on the right, has been adapted from the publication “Optical Terms,” published by Fujinon. The same definition appears in “Measurement and analysis of the performance of film and television camera lenses” published by the European Broadcasting Union (EBU).

TV. Distortion (%) =

$$\frac{\Delta H}{H} \times 100$$

At Imatest we prefer the SMIA definition, which has been widely adopted in the mobile imaging industry, because it is self-consistent. In the traditional definition, TV distortion is the change (Δ) of the center-to-top distance divided by the bottom-to-top distance. In the SMIA definition, both A and B are bottom-to-top distances.

However the traditional TV distortion is likely to be included in an upcoming ISO standard. When this happens we’ll offer a choice (a checkbox).

Although any number in this list can be used as a summary measure of distortion, SMIA TV distortion may be the best choice because it's the easiest to visualize.

- **Coefficient k_1** from the equation, $r_u = r_d + k_1 r_d^3$ where r is normalized to the center-to-corner distance. $k_1 = 0$ for no distortion; $k_1 < 0$ for pincushion distortion; $k_1 > 0$ for barrel distortion.
- **Coefficients h_1 and h_2** from the fifth-order equation, $r_u = r_d + h_1 r_d^3 + h_2 r_d^5$. The selected area must contain at least five horizontal and vertical lines.
- **The Lens Distortion correction coefficient and scale factor** for [Picture Window Pro](#). The sign is the same as k_1 . The scale factor is the value that includes as much as possible of the original image without including areas outside the image. It is less than 1 for barrel distortion and greater than 1 for pincushion.
- **The calculation used for plotting the corrected lines.** Selected in the [input dialog box](#). 3rd order, 5th order, and Picture Window Pro are the choices.

[Picture Window Pro](#) uses a tangent/arctangent model of distortion, which works well for a variety of lenses, including fisheyes.

$$r_u = \tan(10 p_1 r_d) / (10 p_1) ; \quad h_1 > 0 \quad (\text{barrel distortion})$$

$$r_u = \tan^{-1}(10 p_1 r_d) / (10 p_1) ; \quad h_1 < 0 \quad (\text{pincushion distortion})$$

Where p_1 is the correction coefficient. Using Taylor series, we can show that it is similar to the third-order model,

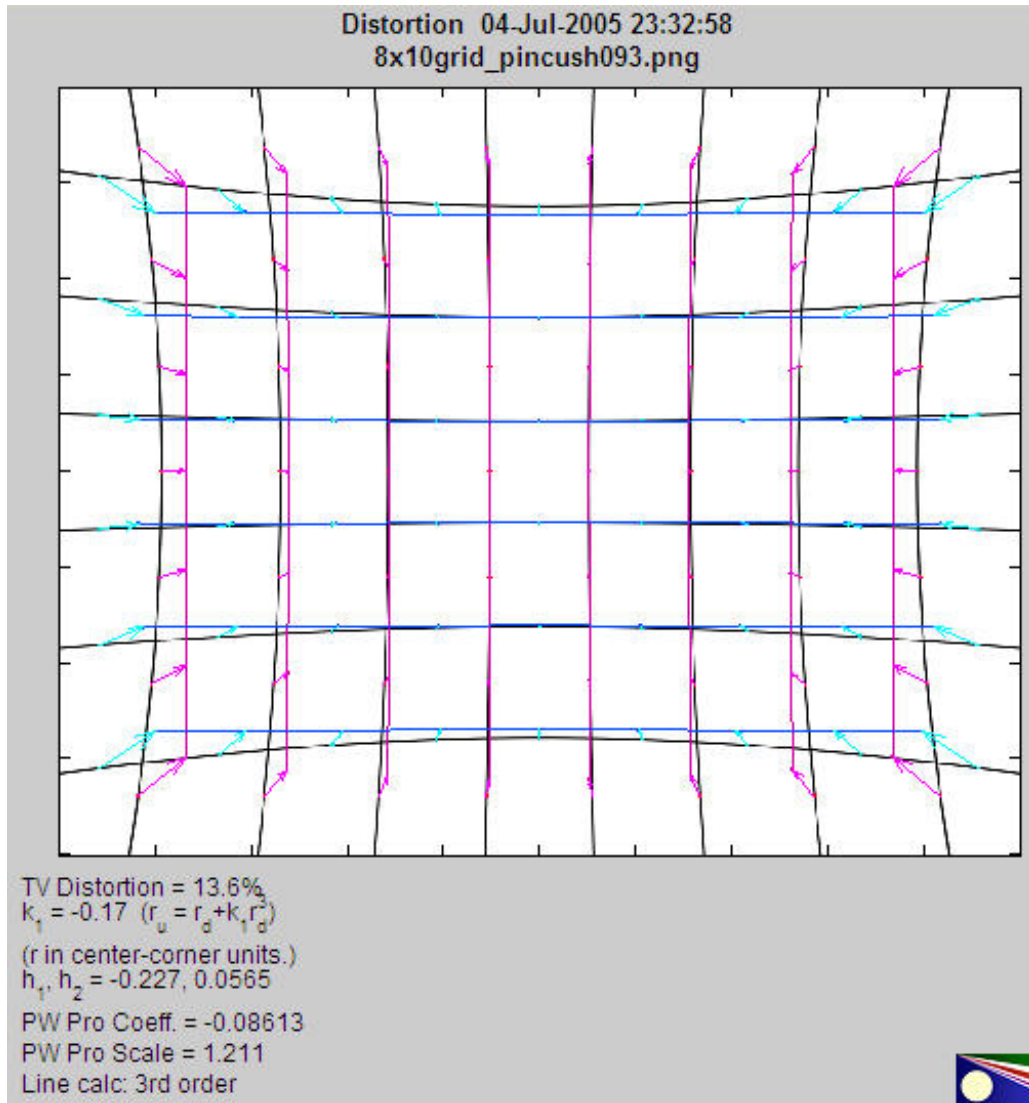
$$\tan(x) = x + x^3/3 + 2x^5/15 + \dots ; \quad \tan^{-1}(x) = x - x^3/3 + x^5/5 - \dots \quad (x^2 < 1) ;$$

$$r_u = x + 100 p_1^2 x^3/3 + \dots (\text{barrel}); \quad r_u = x - 100 p_1^2 x^3/3 + \dots (\text{pincushion}),$$

*$k_1 \approx \text{sign}(p_1) * 100 p_1^2/3$ for small values of k_1 and p_1 . k_1 and p_1 diverge for large values.*

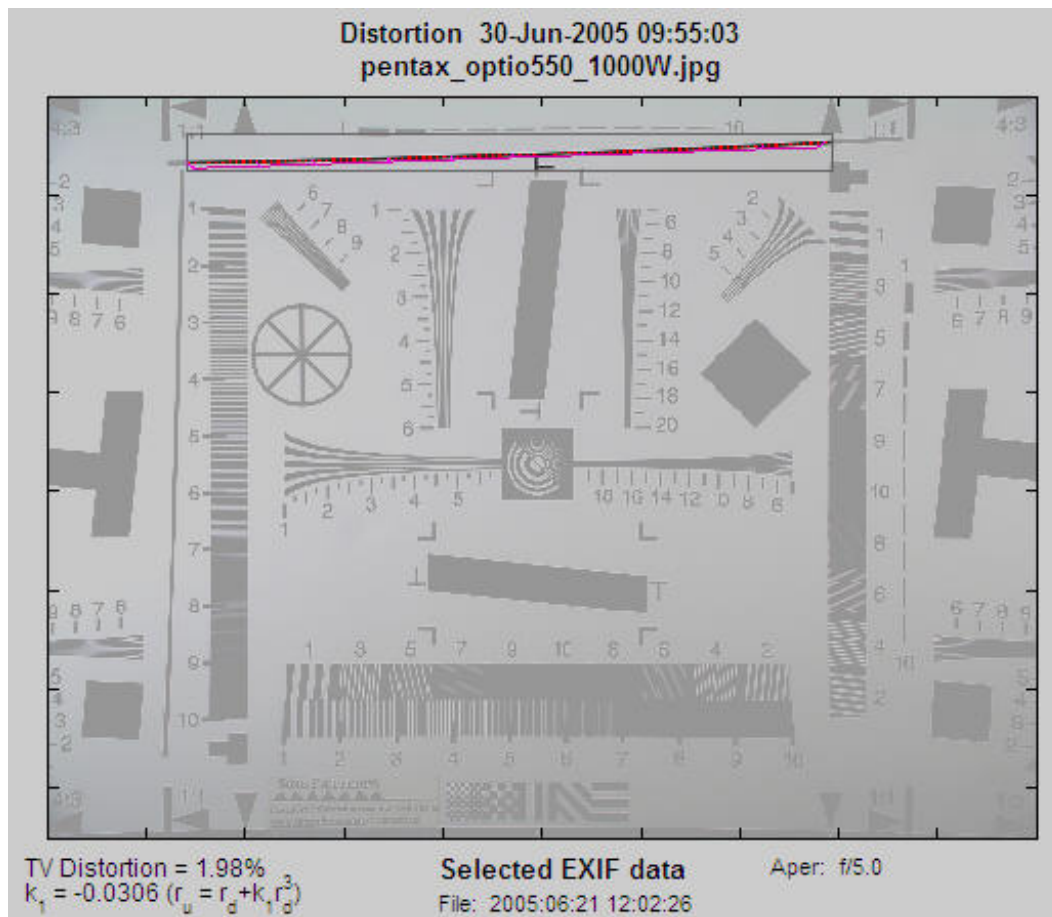
The plot includes arrows that illustrate the change in radius when distortion is corrected. Distortion was too low on the above plot to make the arrows visible. They are illustrated in the plot below for

a large amount of simulated barrel distortion. You can try different line display calculations to see the difference.



Main figure with extreme (simulated) pincushion distortion, illustrating arrows.

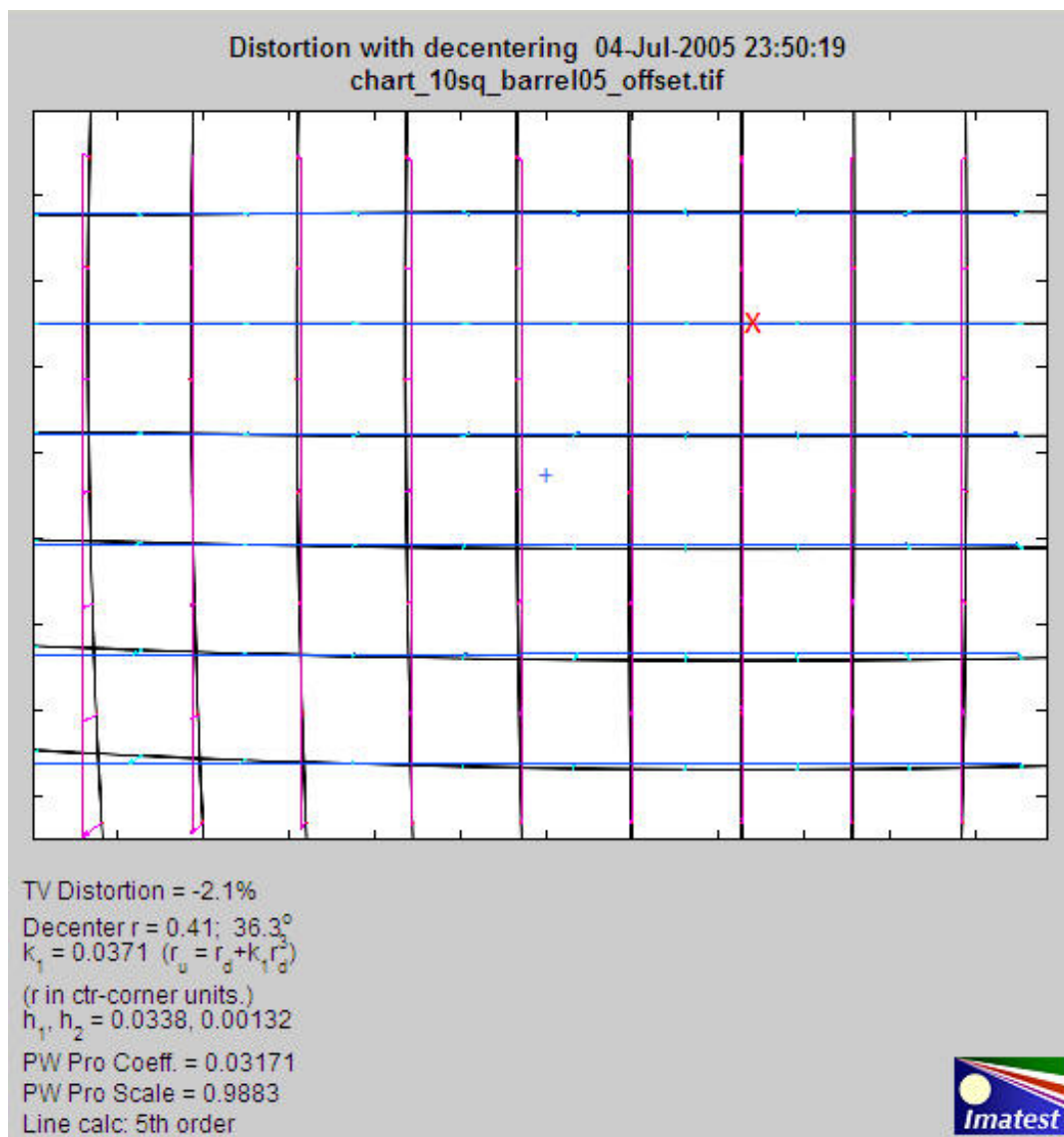
Here is an example of results from an ISO 12233 test pattern, which contains two lines suitable for measuring distortion. They work but they're not ideal: they would be better if they were thinner and closer to the image boundaries. This camera has a modest amount of pincushion distortion. A zoom of a portion of the selected area is shown [above](#).



Decentering (Imatest Master only)

Distortion is normally centered around the geometric center of the image, but it may be decentered due to poor lens manufacturing quality or shock (i.e., dropping the lens). Decentering can appear as a shift in the center of distortion symmetry or as asymmetrical MTF (sharpness) measurements in [SFR](#). Distortion calculates decentering if Decentering calculation is checked in the Input dialog box *and* $|k_1| > 0.01$. (k_1 is the third-order coefficient.) If $|k_1|$ is smaller, distortion is insignificant (and difficult to see); hence decentering has no meaning.

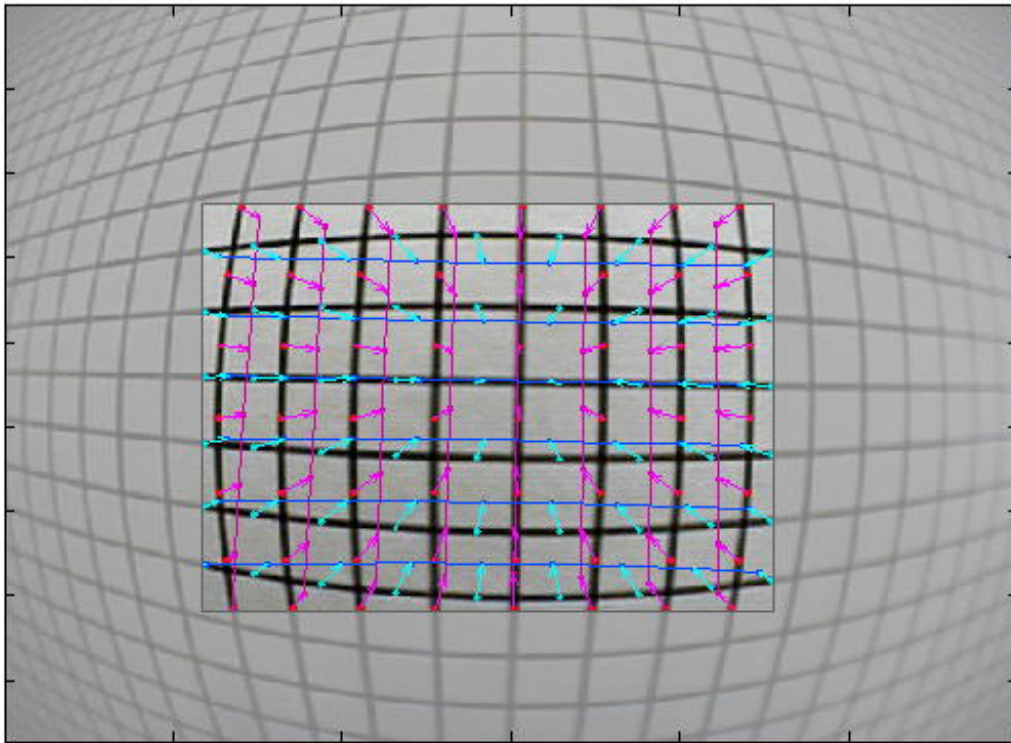
Decentering is reported by the radius (in units of the distance from the image center to the corner) and angle in degrees of the center of distortion symmetry. It is illustrated in the simulated pattern below. The geometrical center of the image is indicated by a pale blue +. The center of the decentered distortion is indicated by **bold red X**.



Results showing centering (+ for image; X for distortion center)

Severe distortion: Corrected image figure

Images from optical systems with severe distortion (fisheye lenses, etc.) often need to be cropped so Distortion can detect vertical and horizontal lines. (See good/bad images, [above](#).) The figure below illustrates an inexpensive fisheye lens. The crop area, shown in the middle, is displayed with full contrast; the area outside the crop has reduced contrast.



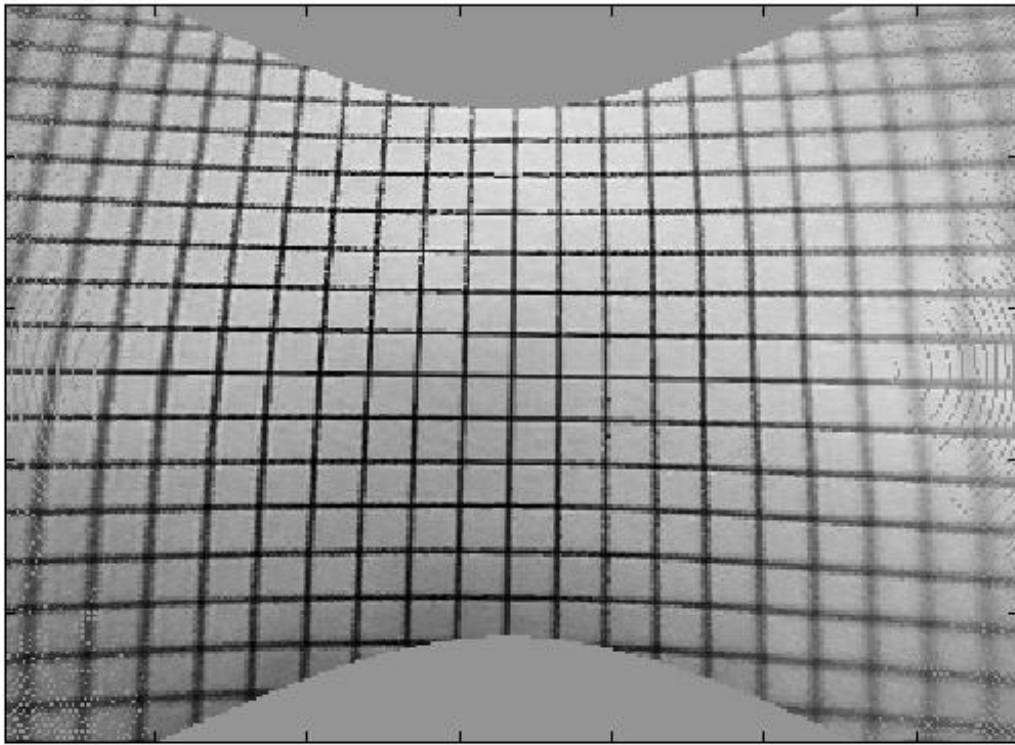
Severely distorted image

This figure does not show the corrected grid image *outside* the crop. The different correction formulas (3rd order polynomial, 5th order polynomial, or tangent/arctangent) can only be compared inside the crop area, where optimum coefficients have been calculated. They don't look very different in this region.

If Display corrected image in the [input dialog box](#) (Imatest Master only) is set to crop only or Always, the corrected image (shown below) is displayed. (The extreme corners are omitted for large amounts of barrel distortion). This figure is not perfect. It is calculated using a simple, fast algorithm that omits some pixels on the left and right. A semicircular "fingerprint" pattern appears in their place. But is good enough to clearly illustrate the performance of the correction algorithm. (Another option, always – interpolated (SLOW!), produces a fine image without any gaps, but is too slow to be recommended.)

In this case, some residual barrel distortion is visible in vertical grid lines on the left and right and some residual pincushion distortion is visible in the horizontal grid lines on the top and bottom. The average correction is quite good. The tangent/arctangent algorithm (used in [Picture Window Pro](#)) gives slightly better results for this image than the 5th order polynomial (which is normally most accurate), and much better than the 3rd order polynomial.

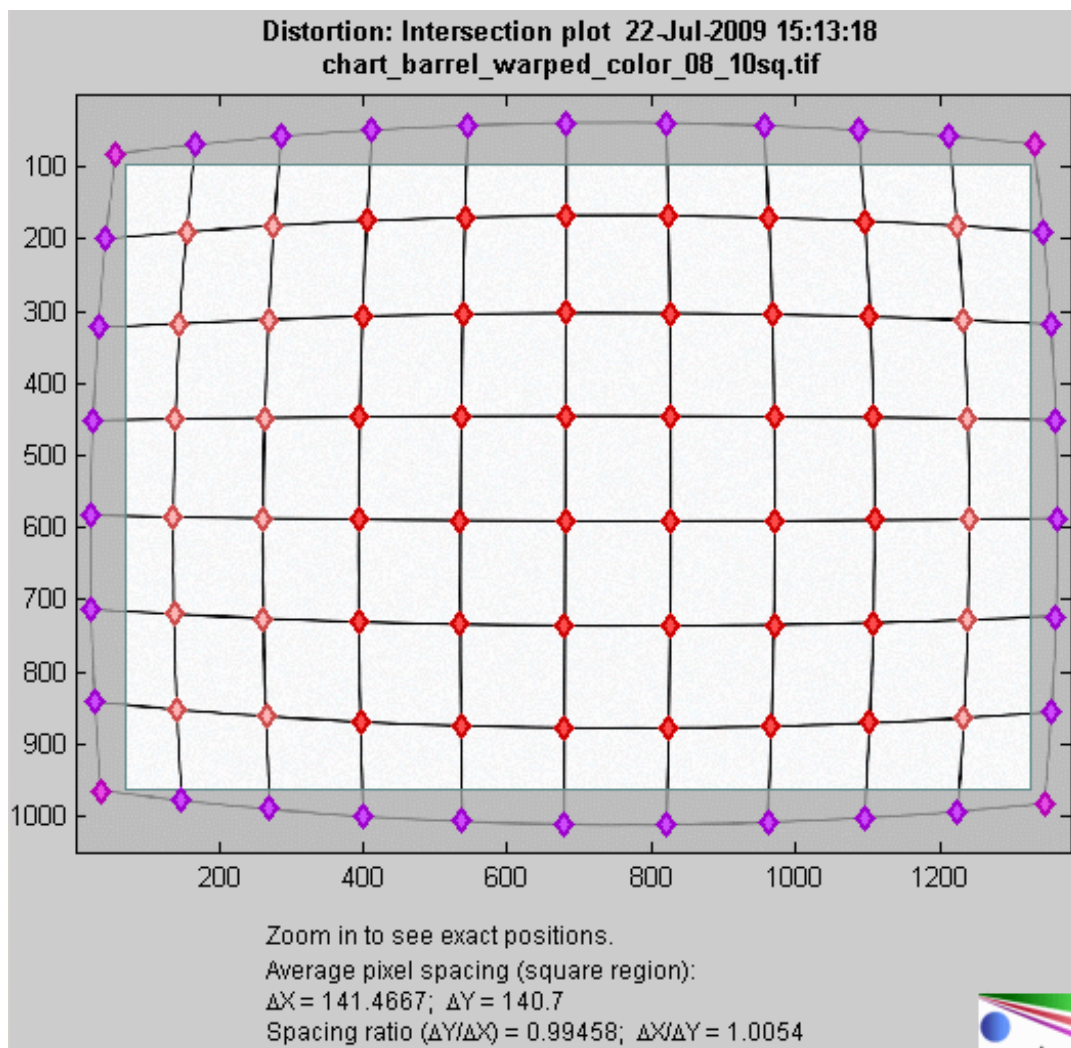
Corrected image: PW Pro 06-Sep-2005 22:37:04
fisheye_grid.jpg



This image has some *perspective distortion*: vertical convergence of lines. This results from the way the camera was pointed when the image was captured. It is not a lens aberration.

Intersection figure (Imatest Master only)

Checking Plot intersection points in the [input dialog box](#) displays a second figure that contains intersection points (line crossings) designated by “+,” “T,” or “L.” “+” intersection points are inside the pattern; “T” points are on the boundaries (top, bottom, and sides), and “L” points are on the corners.



Intersection figure

The image should be cropped so that only “+” intersection points (line crossings) are present inside the crop. Imatest doesn’t like boundaries with “T” or “L” patterns inside the crop. To detect such patterns, they must be outside the crop (as shown above) and Detect points outside crop must be checked in the [input dialog box](#). These points are not used in the for calculating distortion coefficients.

The mean horizontal and vertical line spacings (ΔX and ΔY) are shown beneath the plot and in the optional CSV and XML files, along with the spacing ratios (the distortion aspect ratios), $\Delta Y/\Delta X$ and $\Delta X/\Delta Y$.

If the image contains letters or material unrelated to the the pattern, it should be edited out using an image editor. [Picture Window Pro](#) has a particularly convenient means of creating a rectangular mask for this purpose.

The (x,y) coordinates of the displayed points are included in the optional .CSV and XML output files. These coordinates can be used for further analysis of geometric distortion.

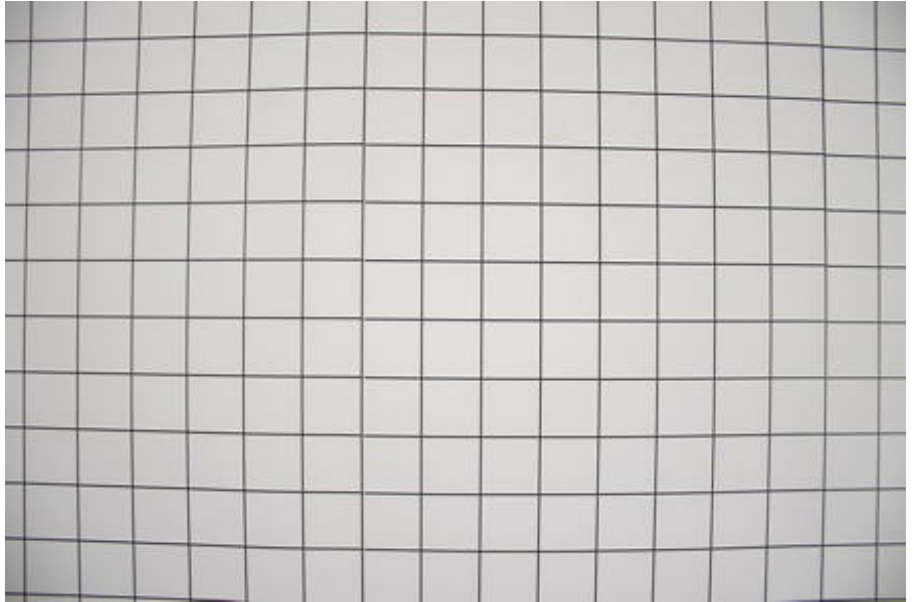
Thanks is due to Karl-Magnus Drake of the [National Archives of Sweden](#) for suggesting this figure.

Radius correction figure (Imatest Master only)

Checking Plot radius correction in the [input dialog box](#) displays a figure that contains detailed distortion/radius correction information, which can be useful for analyzing images like the one on the right, which has barrel distortion near the center but tends towards pincushion near the corners.

Image showing complex (“wave”) distortion: barrel near center; pincushion near corners.

The upper plot shows the distortion-caused change in radius Δr (normalized to the center-to-corner distance, i.e., the half-diagonal) as a function of the distorted (input) radius $r(\text{distorted}) = r_d$.



$$\Delta r = r(\text{corrected}) - r(\text{distorted}) = r_u - r_d$$

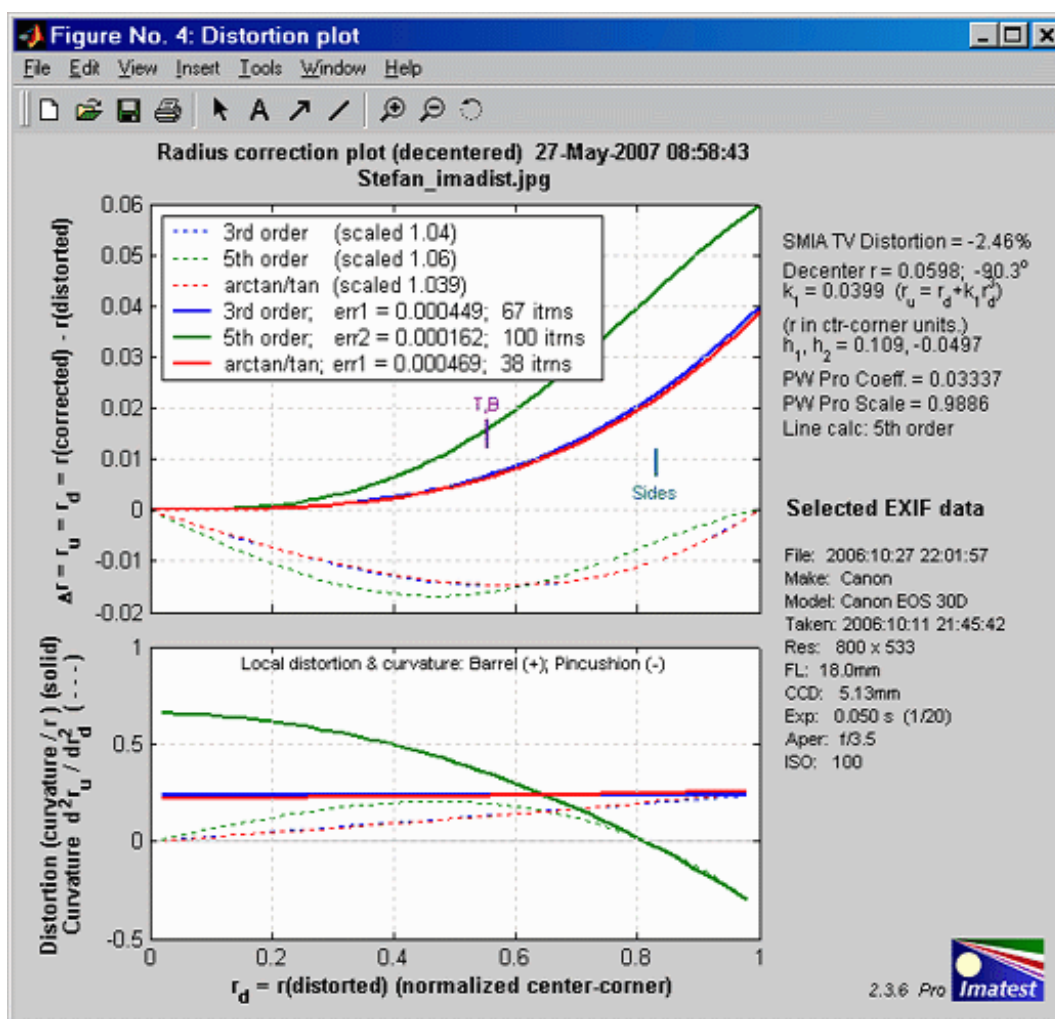
where $r(\text{corrected}) = r_u$ is the corrected (*undistorted*) radius. Δr is a function of the **scaling** (i.e., magnification) of the corrected image.

The **bold** solid lines — in the upper plot show Δr for no scaling, i.e., for the straight correction formulas: $r_u = r_d + k_1 r_d^3$ (3rd order; **blue**); $r_u = r_d + h_1 r_d^3 + h_2 r_d^5$ (5th order; **green**); or the [arctan/tan equations](#) (**red**). With these equations Δr tends to increase as a function of $r(\text{distorted})$, i.e., it tends to be largest near the image corners.

The dashed lines - - - show Δr when r_u (the corrected image) is scaled for no shift ($\Delta r = 0$) at the image corners ($r_d = 1$). The scaling factor is shown in the Legend box, which also shows the error function (described in the [Algorithm](#); different for the 5th order calculation) and the number of iterations.

The upper plot is difficult to relate to the perceived distortion type (pincushion or barrel); the only difference between the solid and dashed lines is the scaling factor (image magnification). The

lower plot, which contains Curvature and Distortion measurements (explained in the **green box** below) is directly related to the distortion type .



Equations for distortion and curvature

The standard distortion equations— $r_u = r_d + k_1 r_d^3$ (3rd order), $r_u = r_d + h_1 r_d^3 + h_2 r_d^5$ (5th order), or the [arctan/tan equations](#)—do not give a clear picture of whether distortion takes the barrel or pincushion form. And to complicate matters, some images like the one above have barrel distortion at some radii and pincushion at others.

The key to determining the type of distortion is to recall that radius r_u is the **position** of a point, its **derivative** dr_u/dr_d is the **slope**, and its **second derivative** $d^2 r_u / dr_d^2$ is the **curvature**—**local curvature**, which is a function of radius r closely related to **distortion**.

It is instructive to look at the first and second derivatives of the 3rd and 5th order equations.

$$3\text{rd order: } dr_u/dr_d = 3 k_1 r_d^2 ; \quad d^2 r_u/dr_d^2 = \text{Curvature} = 6 k_1 r_d .$$

$$5\text{th order: } dr_u/dr_d = 3 h_1 r_d^2 + 5 h_2 r_d^4 ; \quad d^2 r_u/dr_d^2 = \text{Curvature} = 6 h_1 r_d + 20 h_2 r_d^3 .$$

The problem with Curvature as defined here is that it tends to be proportional to r (it's *exactly* proportional for the 3rd order equation). To get a more consistent measurement, we define Distortion as Curvature divided by r .

$$\textbf{Distortion} = \textbf{Curvature}/r = 6 k_1 \text{ (3rd order)} = 6 h_1 + 20 h_2 r_d^2 \text{ (5th order)}$$

Distortion is a single number for the 3rd order case. Distortion and Curvature is only visible in lines that have a **tangential** component. Tangential edges are illustrated in the page on [Chromatic Aberration](#).

Curvature ($d^2 r_u/dr_d^2$) and Distortion (Curvature/ r) are plotted as dashed (- - -) and **bold** solid — lines, respectively, in the lower plot.

Interpretation:

Positive Distortion (and Curvature) represents local barrel distortion;
Negative represents local pincushion distortion.

In the example above, distortion goes from barrel to pincushion around $r_d = 0.8$ (the diameter at the sides of the image). This is visible near the corners, where the (barrel) curved lines straighten out.

The **blue** lines (3rd order equation) are the simplest but least accurate approximation to distortion. The 5th order equation (**red** lines), which has two parameters (h_1 and h_2), is more accurate. If the 3rd and 5th order curves are close, the 3rd order curve is sufficient. In the above example, which is not typically, they are very different; the third order equation is completely inadequate. The arctan/tan equation is also characterized by a single parameter. It behaves differently from the 3rd order equation for large Δr .

The 3rd order distortion value in the figure is a constant solid blue **blue** equal to $6 k_1$.

Links

[Real-Time Lens Distortion Correction](#) by Michael R. Bax (of Stanford) contains the standard

distortion equation, $r_u = r_d (1 + k_1 r_d^2)$ and its inverse (more complex).

[Distortion](#) by Paul van Walree, who also has excellent descriptions of several of the [lens \(Seidel\) aberrations](#) and other sources of optical degradation.

[Lens aberrations](#), including [distortion](#), are discussed in [pages](#) from James R. Graham of the U.C. Berkeley Astronomy Department, based on [Applied Optics & Optical Engineering, Vol XI](#) by J. C. Wyant & K. Creath, which goes into real depth. **Worth checking out if you're mathematically inclined.**

[Optical Metrology Center](#) A UK consulting firm specializing in photographic metrology. (Site must be viewed with Internet Explorer.) They have [a large collection of interesting technical papers](#) emphasizing distortion and 3D applications, for example, [Extracting high precision information from CCD images](#).

[PTlens](#) is an excellent program for correcting distortion. It uses the [equation](#) from [Correcting Barrel Distortion](#) by Helmut Dersch, creator of [Panorama Tools](#), who states,

[Photogrammetry](#) is the science of making geometrical measurements from images. [George Karras](#) has sent me some interesting links with material relevant to latest development:

http://www.vision.caltech.edu/bouguetj/calib_doc/ | <http://nickerson.icomos.org/asrix/index.html>.

The correcting function is a third order polynomial. It relates the distance of a pixel from the center of the source image (r_{src}) to the corresponding distance in the corrected image (r_{dest}):

$$r_{src} = (a * r_{dest}^3 + b * r_{dest}^2 + c * r_{dest} + d) * r_{dest}$$

The parameter d describes the linear scaling of the image. Using d=1, and a=b=c=0 leaves the image as it is. Choosing other d-values scales the image by that amount. a,b and c distort the image. Using negative values shifts distant points away from the center. This counteracts barrel distortion... The internal unit used for r_{src} and r_{dest} is the smaller of the two image lengths divided by 2.

This equation drives me nuts! r_{src} and r_{dest} are reversed from the equations on this page; it only goes up to fourth power (when you include r_{dest} outside the parentheses; third power if you don't); and it has even order terms (a and c) when the theory implies that distortion can be

modeled with odd terms only. Give me a higher order term ($h * r_{\text{dest}}^4$) and dump $a * r_{\text{dest}}^3$ and $c * r_{\text{dest}}$. But nonetheless it works pretty well.

[Mike Collins](#) has posted a technique for finding Ptlens parameters on a [dpreview.com forum](#). I haven't checked to verify it. Briefly,

1) take a shot of the grid pattern and run Imatest distortion. Note the k1 (which is the r^3) parameter.

2) divide my magic number = -3.397. This result is now used as the b (r^3) parameter in Ptlens.

3) Edit the profile file used in Ptlens to include this new lens. Use the bodies multiplier, lens name, focal length, a, b, c (make a and c 0.000000; and b from calculation in 2)). More details on the [post](#).

[Modeling distortion of super-wide-angle lenses for architectural and archaeological applications](#) by G. E. Karras, G. Mountrakis, P. Patias, E. Pets

Algorithm for calculating correction coefficients.

The coefficients of the standard distortion equations are calculated using **nonlinear optimization** that straightens the lines (minimizes their curvature). The equations are

$r_u = r_d + k_1 r_d^3$ (3rd order) r_d is the distorted (input) radius; r_u is the undistorted (output) radius.

$$r_u = r_d + h_1 r_d^3 + h_2 r_d^5 \quad (5\text{th order})$$

$$r_u = \tan(10 p_1 r_d) / (10 p_1); \quad h_1 > 0 \quad (\arctan/\tan; \text{barrel distortion})$$

$$r_u = \tan^{-1}(10 p_1 r_d) / (10 p_1); \quad h_1 < 0 \quad (\arctan/\tan; \text{pincushion distortion})$$

The optimizer

- finds the average locations of vertical and horizontal lines in the image,
- scans *between* the lines to find (x,y) locations on the lines (and if there is room, scans lines that intersect the image boundaries),

- finds polynomial fit for each line,
- minimizes the sum of squares of the second order polynomial coefficients, which are the line curvatures. For the 5th order case the optimizer minimizes the sum of squares of the second and fourth order polynomial coefficients— a slightly different ***error function*** from the 3rd order and arctan/tan cases.

Since optimization is performed only to straighten curved lines, this algorithm is relatively insensitive to perspective distortion and small amounts of camera misalignment.
