

disto - User Guide

Version 1.0.2 - © 2026 Marco Ristuccia



Contents

1	What disto does	3
2	Installing	3
2.1	macOS (Apple silicon)	4
2.2	Windows (x86, 64-bit)	4
3	Before you start	4
3.1	Export the full frame	5
3.2	Shifts: enter the direction the picture looks	5
3.3	Pixel pitch and the camera list	6
4	The desktop app	7
4.1	The window	7
4.2	Lists of files and their shifts	8
4.3	Geometry options	8
4.4	When a camera is missing	9
4.5	The Calibrate tab	10
4.6	The Inspect tab	14
4.6.1	A profile's distortion chart	15
4.6.2	Checking an image and a shift	16
4.7	The Correct tab	16
4.8	The Cameras tab	19
5	Workflows	20
5.1	Shooting a calibration series	20
5.2	Correcting a shoot	21
5.3	Stitching shifted frames	21
6	Troubleshooting	22
7	Command-line reference	23
7.1	calibrate	23
7.2	correct	24
7.3	inspect	25
7.4	cameras	25
7.5	The profile file	26
7.6	Exit codes	27

1 What disto does

Almost every lens bends straight lines a little. Lines near the edges of the picture bow outwards (*barrel* distortion) or inwards (*pincushion* distortion), and some wide-angle lenses do a bit of both, bowing one way near the centre and the other near the edge, a shape often called “moustache” distortion. Whatever its shape, the amount depends only on how far a point lies from the lens’s *optical axis*, the line through the centre of the lens. Points on the axis are not moved at all; the further out a point lies, the more it is moved.

Mainstream lens profiles, such as those built into RAW developers, assume that the optical axis passes through the centre of the sensor. On an ordinary camera that is true. On a tech camera with a shifted lens it is not: shifting slides the lens sideways, or up and down, over the sensor, so the optical axis lands somewhere else, sometimes even outside the frame. A correction centred on the sensor then straightens the wrong part of the picture and can make a shifted frame worse rather than better.

disto models the *lens* instead of the sensor. A profile describes the lens’s distortion in millimetres around its own optical axis, across the whole *image circle*, the full circle of picture the lens projects, of which the sensor only ever sees a part. When you correct a shot, you tell disto the shift you used, and it works out where the optical axis fell on that particular frame and corrects around that point, even when the point lies outside the frame.

The work falls into two steps:

1. **Calibrate once per lens.** Photograph a scene full of long straight edges (a brick wall or a building façade works well) at several shift positions. disto finds the edges, measures how the lens has bent them, and calculates the distortion coefficients that make them straight again, across all frames at once. Because each frame sees the lens from a different place on the image circle, together they cover far more of it than any single frame could. The result is a *profile*: a small file describing that lens.
2. **Correct every shot.** Give disto the profile and the shift of each shot. For each file, it works out where the optical axis falls on that frame and resamples the image so that straight lines become straight.

disto never reads RAW files. Your RAW developer stays in charge of colour, tone and everything else that makes the picture look the way you want; disto only changes the geometry of the exported TIFF, and passes everything else through untouched. The ICC colour profile and the EXIF and XMP metadata of each input are carried over into the corrected file. Profiles are small JSON files, with all distances in millimetres. Because they do not depend on pixel counts, the same profile serves a full-resolution export and a downsampled one.

disto comes as a desktop app (**Disto.app**) and as a command-line tool. The app is a friendly front end for the tool: each form builds a command line and runs it through the same code, so both always give identical results. The app also shows you every command it runs, so you can copy it into a script later if you want to automate a task.

2 Installing

disto is available for Macs with Apple silicon and for 64-bit (x86) Windows.

2.1 macOS (Apple silicon)

The desktop app. Open the disk image (`Disto-version.dmg`) and drag `Disto.app` to Applications.

The app is signed with an Apple Developer ID and *notarised*: Apple has checked it for malicious software and vouched for it, and macOS verifies this the first time the app opens. Because disto is not distributed through the Mac App Store, macOS only lets it open if your Mac accepts apps from identified developers as well as from the App Store. If macOS refuses to open the app, open System Settings → Privacy & Security, and under Security set Allow applications from to App Store & Known Developers. The first time you open disto, macOS reminds you that it was downloaded from the internet; click Open. It does not ask again.

The command-line tool. The tool is already inside the app and uses the app's runtime. If you want to type `disto` in a terminal, link it once into a folder on your path:

```
| ln -s /Applications/Disto.app/Contents/MacOS/disto-cli /usr/local/bin/disto
```

2.2 Windows (x86, 64-bit)

The desktop app. Run the installer (`Disto-version-setup.exe`). It installs disto for all users in `C:\Program Files\Disto`, adds Disto to the Start menu, and adds an entry under Settings → Apps from which you can remove it again. Installing needs administrator rights, so Windows asks you to confirm before the installer starts. To update disto, run the installer of the new version: it replaces the previous one. On Windows on Arm the x86 version runs under Windows' built-in emulation.

Note. I do not own a Windows code-signing certificate, so the installer is not signed. Windows therefore cannot tell who published it and warns you before running it. If you want to install disto, you have to accept that risk and tell Windows to go ahead anyway:

- Your browser may flag the download as uncommon or unsafe. Choose to keep it (in Microsoft Edge: ... → Keep, then Show more → Keep anyway).
- When you run the installer, Microsoft Defender SmartScreen shows Windows protected your PC. Click More info, check that the file is the one you downloaded, then click Run anyway.
- The User Account Control prompt that follows shows the publisher as Unknown. Click Yes to install.

The command-line tool. The installer puts the tool next to the app, as `disto-cli.exe` in `C:\Program Files\Disto`; it uses the app's runtime. It is not added to the PATH, so either type its full path:

```
| "C:\Program Files\Disto\disto-cli.exe" --help
```

or add `C:\Program Files\Disto` to your PATH once (Settings → System → About → Advanced system settings → Environment Variables), after which `disto-cli` works in any new terminal. Wherever this guide writes `disto` at the start of a command, type `disto-cli` on Windows.

3 Before you start

A correction can only be as good as the information disto is given. Three things have to be right for every file: the exported file itself, the shift you enter, and the *pixel pitch* of the sensor (the

size of one pixel, which links the millimetres of the profile to the pixels of the image). `disto` checks whatever it can, but some things are invisible to it: it cannot see how a file was exported, nor what the shift scale on your camera read. This section explains each of the three, and why it matters.

3.1 Export the full frame

`disto` works out where the optical axis lies by measuring from the centre of the frame the sensor recorded. Anything that changes that frame's geometry before `disto` sees it moves the centre, and with it the axis. So every TIFF you give `disto` must be the whole, uncropped frame, exactly as the sensor recorded it:

- **No crop, rotation, keystone or perspective correction.** Each of these moves pixels away from where the sensor put them. Do them after `disto`, on the corrected file, if you need them.
- **No distortion or chromatic-aberration correction in the RAW developer.** Otherwise the file is corrected twice, once by a profile that assumes an unshifted lens and once by `disto`.
- **Downscaling is fine**, as long as the aspect ratio stays the same. `disto` recognises a smaller export and scales its measurements to match.
- **16-bit is strongly preferred.** 8-bit files work, but `disto` warns you, because resampling can show banding in smooth gradients such as skies.
- **Output sharpening off.** Correction resamples every pixel, so sharpening is best applied to the final, corrected picture.

As a safeguard, `disto` compares each file's aspect ratio with the sensor's and refuses a file that looks cropped. If you are sure the file is not cropped, you can tell it to continue anyway (§6).

3.2 Shifts: enter the direction the picture looks

For every file you give `disto`, you also give the shift used for it: two numbers in millimetres, as read off the camera's shift scale. The first, s_x , is the lateral (left–right) shift; the second, s_y , is the rise or fall.

The sign of each number follows one simple rule: it is the direction the *picture* looks. Shifting the lens to the right makes the picture look further to the right; raising the lens makes it look further up. So, for a lens that moves, the sign is simply the direction of the lens movement:

Value	Lens movement (seen from behind the camera)	Picture looks
$s_x > 0$	shifted right	right
$s_x < 0$	shifted left	left
$s_y > 0$	rise	up
$s_y < 0$	fall	down

For example, a frame taken with the lens raised by 15 mm and shifted 10 mm to the left is entered as $s_x = -10$, $s_y = 15$.

Some cameras move the *rear standard* (the back) instead of the lens, on one axis or on both. Moving the back has the opposite effect on the picture: lowering the back by 18 mm makes the picture look up, exactly as raising the lens by 18 mm would. You do not have to invert these

readings in your head. Instead, tell disto which axes you read from the rear standard (the **From the rear standard** boxes in the app, `--rear` on the command line) and type the readings exactly as the scale shows them. disto inverts those axes for you:

App	Command line	Inverts
Lateral (x) ticked	<code>--rear x</code>	lateral readings
Rise/fall (y) ticked	<code>--rear y</code>	rise/fall readings
both ticked	<code>--rear xy</code>	both

On a Cambo-style body, for instance, lateral shift is on the front standard and rise/fall is on the rear, so you would tick Rise/fall (y) only. Check once which way the picture moves on your camera, then use the same setting for every shot, both when you calibrate and when you correct.

Note. A shift entered wrongly costs about as much accuracy as a wrong sign: in both cases disto corrects around the wrong point. Before correcting a batch, it is worth checking one shift entry on the Inspect tab (§4.6.2).

3.3 Pixel pitch and the camera list

Profiles measure distances in millimetres on the sensor; images measure them in pixels. To go from one to the other, disto needs the size of one pixel on the sensor, called the *pixel pitch*. For example, the Hasselblad CFV 100C has a sensor 43.8mm wide with 11 656 pixels across, so its pixel pitch is $43.8/11656 \approx 0.00376$ mm, or 3.76 μm .

You normally never have to type a pitch. disto reads the camera’s make and model from the EXIF tags in each TIFF and looks them up in its *camera list*, which records each camera’s sensor size in millimetres and its pixel count. The list comes with common medium-format backs, and you can add your own (§4.8). A model name followed by a qualifier, such as CFV 100C/Electronic Shutter as written by Phocus, still matches its entry.

Knowing the sensor’s full pixel count also lets disto notice a downsampled export. If a CFV 100C file is only 5828 pixels wide, half the native width, each of its pixels covers twice as much of the sensor, and disto doubles the pitch to match. You do not need to do anything for this to happen. When a file’s camera is not in the list, or the file carries no camera tags at all, disto cannot find the pitch by itself. You can then give it for that run, in one of two ways:

Pixel pitch in millimetres (so 3.76 μm is entered as 0.00376). This is the pitch of the pixels in *this file*: for a full-resolution export it is the sensor’s pitch, but for an export downsampled to half size you must double it yourself, because without a camera entry disto cannot tell that the file was downsampled.

Sensor width in millimetres. disto then divides it by the file’s width in pixels. That is only right for a full-resolution export, since a downsampled file has fewer pixels across the same sensor.

Both of these apply only to the current run. Adding the camera to the list is usually the better fix: you do it once, it works for every later file, and disto then handles downsampled exports for you.

4 The desktop app

4.1 The window

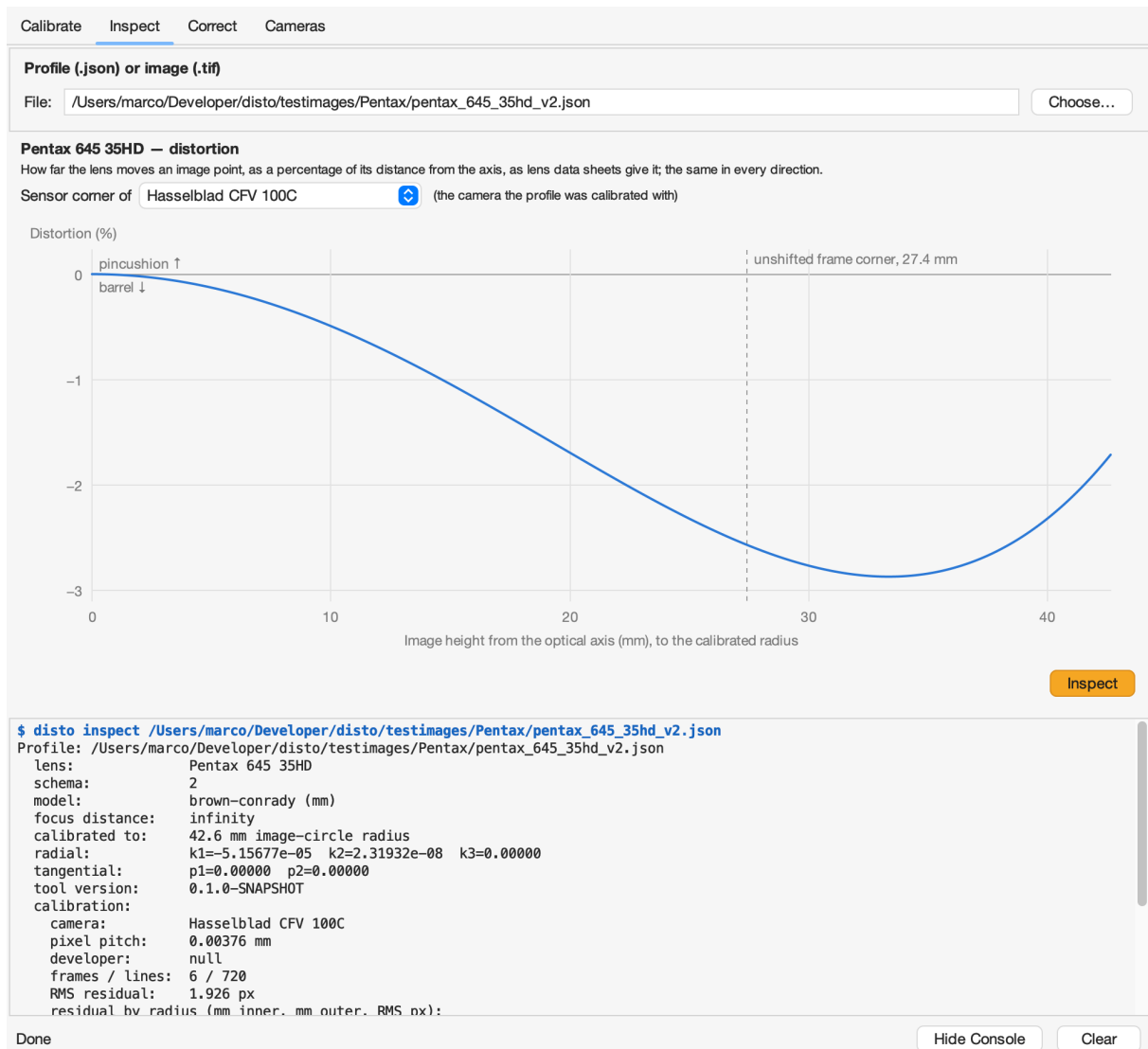


Figure 1: The Disto window: a tab for each task above the output console.

The window has four tabs, arranged in the order of the work: **Calibrate**, **Inspect**, **Correct** and **Cameras**. Below them sits an output console, where **disto** writes its reports (Figure 1). The app opens on **Correct**, since that is the task you will do most often.

- **Running a task.** The first three tabs are forms. When you press the orange button at the bottom right of one (**Calibrate**, **Inspect** or **Correct**), **disto** checks the form, builds the matching command line and runs it. The console first echoes the command, in bold blue, then shows the report as it arrives, with errors in red. You could type the same command in a terminal and get exactly the same result.
- **Mistakes in a form**, such as a missing lens name, are caught before anything runs and explained in a dialog, so you can fix them and press the button again.
- **The status bar** at the bottom tells you what is running, and how the last run ended: **Done**, **Done** — 2 warnings in the console, or **Failed** with its exit code. While a command

runs, the run buttons are disabled, so you cannot start a second task by accident. Clear empties the console.

- **Showing and hiding the console.** Hide Console in the status bar, or View → Show Console (Command-L), hides the console and gives its room to the tabs; this is handy on the Inspect tab, where the chart benefits from the space. The console starts hidden, with the window tall enough to show the whole Calibrate tab. It opens by itself whenever a command runs, so you never miss a report or an error.
- **Command-line help.** Help → Command-Line Reference prints the usage of every command into the console.
- **Appearance.** The app follows the system's light or dark appearance, and switches when the system does.

4.2 Lists of files and their shifts

The Calibrate and Correct tabs each hold a list of TIFF files, with the shift of each one next to it (Figures 3 and 5). The list works the same way on both tabs.

- **Adding files.** Drag TIFFs from the Finder onto the list, or press Add... and choose them. Each new file takes the shift currently in the Shift *sx* and *sy* fields beside the buttons. So when you add a group of shots all taken at the same shift, type that shift into the fields first, then add the files, and they arrive with the right values. Files that are already in the list, and files that are not TIFFs, are skipped.
- **Editing shifts.** Double-click a shift in the list to change it. You can type a comma or a point as the decimal separator. To change several rows at once, select them, type the shift into the fields and press Set for selected. With no rows selected, Set for selected applies the shift to every row.
- **Importing shifts.** If you already keep your shifts in a spreadsheet or text file, Import shifts CSV... reads them from rows of `file,sx,sy`. A row naming a file that is already in the list (by file name) sets that file's shift. A row naming any other file adds that file to the list, looked up relative to the folder the CSV is in. A header row, and lines starting with #, are ignored. If the CSV names files that cannot be found, a warning lists them.
- **Removing files.** Select the rows and press Remove. This only takes them off the list; the files themselves are not touched.

Above each list, the From the rear standard boxes say which axes you read from the rear standard (§3.2). The heading of the list spells out what the current setting means in practice. With Rise/fall (y) ticked, for example, it reads *sx > 0 looks right, sy < 0 looks up*, a handy reminder of which way round to type the numbers.

4.3 Geometry options

The Geometry section of the Calibrate, Inspect and Correct tabs holds three settings. Each applies to every file in the run, and in most cases you can leave all three as they are.

Orientation How the sensor is turned relative to the lens's shift movements. If you rotate the back for a portrait frame, the picture's horizontal and vertical no longer line up with the lens's lateral and rise/fall movements, and disto needs to know by how much the back was turned to place the optical axis correctly. The default, `auto`, reads this from the file's EXIF

orientation tag, which is right for most files. Choose 0, 90, 180 or 270 degrees only if the tag is missing or wrong.

Pixel pitch (mm) and **Sensor width (mm)** Greyed out unless every listed file lacks camera tags; on the Inspect tab, unless the chosen image lacks them. As soon as one file names its camera, disto takes that camera for the whole batch and finds the pitch in the camera list, so a missing camera is added to the list (§4.4) rather than typed in here. When the fields are available, fill in one of them (not both), as explained in §3.3.

4.4 When a camera is missing

As soon as you add files to the Calibrate or Correct tab, disto reads the camera tags of each one and checks them against the camera list. It reads only the tags, not the image itself, so this is quick even for very large files. There are three possible outcomes:

- **The camera is in the list.** All is well, and nothing is shown.
- **The file names a camera that is not in the list.** A dialog offers to add it, with the make, model and pixel size already filled in from the file. Type the sensor's width and height in millimetres, from the camera's specifications; the dialog shows the pixel pitch they give, so you can check it looks sensible. Check the pixel size too: it is taken from the file, so if that file was downsampled or cropped, replace it with the full-resolution size from the specifications. **Add to Camera List** saves the camera permanently. **Not Now** skips it for the moment; disto will not ask again about that camera until you restart the app.
- **The file names no camera at all.** disto cannot look anything up. When no listed file names a camera, the pitch fields become available and a note asks you for a pixel pitch or a sensor width. When other files do name one, the fields stay greyed out and the note asks you to remove the files without camera tags, or to process them in a separate run.

As long as the pitch cannot be found for every file, an orange note stays above the geometry fields (Figure 2). For a camera you skipped, it offers an **Add to Camera List...** button, so you can change your mind. The note disappears once the camera has been added, or, for files that name no camera, once you type a pixel pitch or a sensor width.

Example Back 60 is not in the camera list: add it, or enter the pixel pitch or sensor width below.
1 of 3 files name no camera: enter the pixel pitch or sensor width below.

Add to Camera List...

Figure 2: The note shown when one listed file comes from a camera that is not in the list and another names no camera.

4.5 The Calibrate tab

In short: what this tab does

The Calibrate tab is where disto learns how a lens distorts. You give it a handful of photographs of a scene with long straight edges, such as a brick wall or a building façade, each taken with the lens shifted to a different position, and you tell it the shift of each one.

disto then finds the straight edges in every frame and measures how much the lens has bent them. From all the frames together, it calculates a few numbers, the *distortion coefficients*, that describe the lens’s distortion. Applied in reverse, they make every edge in every frame straight again at once.

The result is saved as a *profile*: a small `.json` file that describes this lens at this focus distance. You calibrate once per lens (and again for each focus distance you work at, if you work close up). After that, you use the profile on the Correct tab for every picture taken with the lens.

Before your first calibration, please read §5.1 on shooting the frames. The choice of scene and the spread of shifts matter far more than any setting on this tab: good frames give a good profile with the default settings, and no setting can rescue poor frames.

The form is filled in from top to bottom.

Lens. These fields describe what you are calibrating. They are stored in the profile so that you, and disto, can later tell what it is for.

Lens name Required. Recorded in the profile and shown wherever the profile is used, so choose a name you will recognise, such as *Rodenstock HR Digaron-W 40*.

Focal length (mm) Optional, recorded for reference only.

Focus distance inf for infinity, or the distance in metres at which the lens was focused. A lens’s distortion changes a little with focus, so a profile is only exact at the distance it was calibrated at. For landscape and architecture, an infinity profile is usually all you need; for close work, calibrate again at a distance close to the one you shoot at.

RAW developer Optional, recorded for reference. A profile is best used on files exported from the same developer, with the same settings, as the frames it was calibrated from; this field reminds you which those were.

Frames. Add the calibration photographs and enter the shift of each one, as described in §4.2. disto needs at least three frames at different shifts; five to nine is better (§5.1). Take particular care with the shifts and the **From the rear standard** boxes: a wrong shift makes the frame’s edges look as if they were bent differently, and spoils the fit for all the others.

Model. This section chooses which coefficients disto calculates. Each one describes a different part of the distortion’s shape. The fit always includes the main term, k_1 , which captures the overall amount of barrel or pincushion distortion. The others are optional:

k2 On by default. It describes how the distortion changes further out on the image circle, including the “moustache” shape of many wide-angle lenses, where the distortion turns from one direction to the other. Untick it only for a lens with very simple distortion.

k3 A further term that matters mostly at the very edge of the image circle. Try it when the residual (see below) climbs towards the edge.

Lens

Lens name: Pentax 645 35HD

Focal length (mm): 35

Focus distance: inf

RAW developer: Phocus

Shift readings

From the rear standard: ☐ Lateral (x) ☒ Rise/fall (y)

Frames — at least three, at different shifts; sx > 0 looks right, sy < 0 looks up

File	sx (mm)	sy (mm)
B0010950_X00_Y00.tif	0	0
B0010949_Xp18_Y00.tif	18	0
B0010951_Xm18_Y00.tif	-18	0
B0010952_Xm18_Ym18.tif	-18	-18
B0010953_X00_Ym18.tif	0	-18
B0010954_Xp18_Ym18.tif	18	-18

Add...

Remove

Import shifts CSV...

Shift sx 0 sy 0 mm

Set for selected

Geometry

Orientation: auto

Pixel pitch (mm):

Sensor width (mm):

Model

☒ k1
☒ k2
Fit: ☐ k3 (lens whose residual climbs at the image-circle edge)
☐ p1, p2 (visibly decentered lens)

Output

Profile: /Users/marco/Profiles/pentax_645_35hd.json

Choose...

☐ Proceed despite too few frames or edges

☒ Verbose report

Calibrate

Figure 3: The Calibrate tab, with six frames of a brick wall shot at shifts of up to 18 mm.

p1, p2 Decentring terms, for a lens whose elements are not quite centred on one another, so that the distortion differs from one side of the picture to the other. Most lenses do not need them.

More terms are not automatically better. A term the lens does not need has nothing real to measure, so it ends up fitting the small irregularities of the scene instead, and makes the profile less reliable. Leave k_3 and p_1 , p_2 off unless a first calibration shows that they are needed; the report helps you decide (see *k3 check* below).

Output. Choose where to save the profile, in *Save profiles as*. Two options change how the run behaves:

Proceed despite too few frames or edges Normally *disto* refuses to save a profile when the frames do not hold enough information for a dependable result (see *When calibration refuses* below). This option overrides those safety checks. Use it only to look at a doubtful result, never for a profile you mean to keep.

Verbose report Adds more detail to the report: where the optical axis fell on each frame, and how the residual varies with the distance from the axis.

Running. Press *Calibrate*. Calibration takes a while: six 100 MP frames take about a minute on a recent Mac. When it finishes, the report appears in the console and the profile is saved.

Reading the report. The report tells you how well the profile fits the frames, and points out anything that deserves a second look. Here is the report of a six-frame calibration of a Pentax 645 35HD on a brick wall, shifted up to 18 mm, slightly shortened:

```
Detecting edges and fitting k1, k2, k3 across 6 frames...

edges traced:      720 (2112340 edge points)
RMS residual:      1.885 px
calibrated to:      42.6 mm image-circle radius
longest edge:       43.8 mm
edge directions:    688 horizontal (9046 mm), 29 vertical (184 mm),
                    3 oblique (16 mm)

k1 = -5.987278e-05
k2 = 3.339436e-08
k3 = -3.677961e-12

k3 check, the same edges fitted with and without k3:
  RMS residual:      1.885 px with k3, 1.924 px without
  outermost band:    1.378 px with k3, 1.593 px without
  corrections differ by at most 3.5 px within 39.7 mm of the axis
  and by 298 px at the farthest frame corner, 52.7 mm out, beyond the
  42.6 mm calibrated
  advice: k3 adds little here; without it the correction stays steadier
  beyond the edges. Where the edges are, k3 changes the correction by 3.5 px
  at most, within the scene's own scatter (twice the RMS, 3.8 px). Either way
  the frame corners beyond the calibrated radius are extrapolated; trust
  neither fit there.

per frame:
  B0010950_X00_Y00.tif      120 lines   rms 2.019 px
  B0010949_Xp18_Y00.tif     120 lines   rms 1.840 px
  ...
```

warning: 98% of the edge length runs horizontally. The radial profile is still measured, but nothing checks it from the other direction; a scene with long vertical edges as well is better.

Line by line:

edges traced How many straight edges disto found and used, across all frames, and how many individual points they are made of.

RMS residual The most important figure. After applying the profile, disto measures how far the points of each edge still are from a perfectly straight line, and averages that distance (as a root mean square) over all edges. It is given in pixels, and lower is better. Do not expect zero, though: real scenes are never perfectly straight. On the wall above, mortar courses that pass right through the optical axis, where distortion cannot bend anything, still bow by 3 to 13 pixels, so about 2 pixels is as low as that wall can go. A residual close to the scene's own irregularity is a good result.

calibrated to How far from the optical axis the frames' edges reach, in millimetres. This is how far out the profile is backed by actual measurements. Correcting a picture whose corners go beyond this radius means guessing (extrapolating) the distortion there, and the Correct tab warns you when that happens. More, and larger, shifts during calibration push this radius further out.

longest edge The longest straight edge found, in millimetres on the sensor. Long edges are what make the measurement reliable.

edge directions How many edges run horizontally, vertically or at an angle, with their total length. When nearly all of the length runs one way, as the courses of a brick wall do, a warning follows. The overall distortion is still measured, but nothing confirms it from the other direction, and the decentring terms p_1 , p_2 cannot be measured at all. A scene with long edges both ways is better.

k_1 , k_2 , ... The calculated coefficients, as stored in the profile. You do not need to interpret them; the Inspect tab turns them into a chart (§4.6).

k3 check To help you decide whether you need k_3 , disto fits the same edges a second time, the other way round: with k_3 if you calibrated without it, without k_3 if you calibrated with it. It then compares the two fits: their residuals, overall and in the outermost band of the image circle, and how much the two corrections differ where the edges actually are. (A uniform change of scale is left out of that comparison, since it makes no difference to whether lines are straight.) When frames reach beyond the calibrated radius, it also gives the difference at the farthest corner. k_3 grows very quickly with the distance from the axis, so out there it dominates the correction, with no measurements to check it.

advice disto's recommendation, with its reason. It advises k_3 only when k_3 changes the correction by more than twice the RMS residual (the scene's own scatter) *and* also fits the outermost edges better. Otherwise the simpler profile is the safer choice, because it behaves more steadily beyond the measured edges. This check is skipped when only k_1 is fitted.

per frame The number of edges and the residual for each frame. A frame whose residual is far above the others is marked <- **check this frame's shift entry**: that nearly always means its shift was entered wrongly. The mark works by comparing the frames with one another, so it cannot catch a mistake repeated on many frames. If the residual is high on every frame, check all the shift entries and the rear-standard boxes.

residual by radius Shown only with Verbose report: the residual for bands at increasing distance from the axis. A residual that grows steadily towards the outer bands suggests

trying k3.

When calibration refuses. Unless **Proceed despite...** is ticked, calibration stops with an explanation, and saves nothing, when:

- there are fewer than three frames;
- fewer than 20 straight edges were found;
- the longest edge spans less than 35% of the frame's shorter side; or
- the fit does not converge, that is, *disto* cannot find a set of coefficients that consistently straightens the edges.

Each of these means the frames do not hold enough information for a dependable profile. Rather than forcing a result, reshoot the frames; §5.1 explains what to change.

4.6 The Inspect tab

In short: what this tab does

The Inspect tab lets you look before you act. It never changes or writes any file; it only shows you information. It does one of two jobs, depending on the file you give it:

A profile (.json): the tab draws a chart of the lens's distortion, the same kind of curve lens makers print in their data sheets. Use it to check that a fresh calibration looks plausible, or to compare lenses.

A TIFF and a shift: the tab reports how *disto* understands that file: which camera and pixel pitch it found, how the file was scaled and oriented, and, most importantly, where it places the optical axis for the shift you entered. This is the quickest way to make sure your shift entries and rear-standard setting are right before you correct a whole batch.

Choose a file with **Choose...**, or drop it onto the field, then press **Inspect**; the description appears in the console. The settings under **For images only** apply only to TIFFs, and are hidden while a profile is chosen, to give the chart more room (Figure 4). For the largest chart, hide the console too.

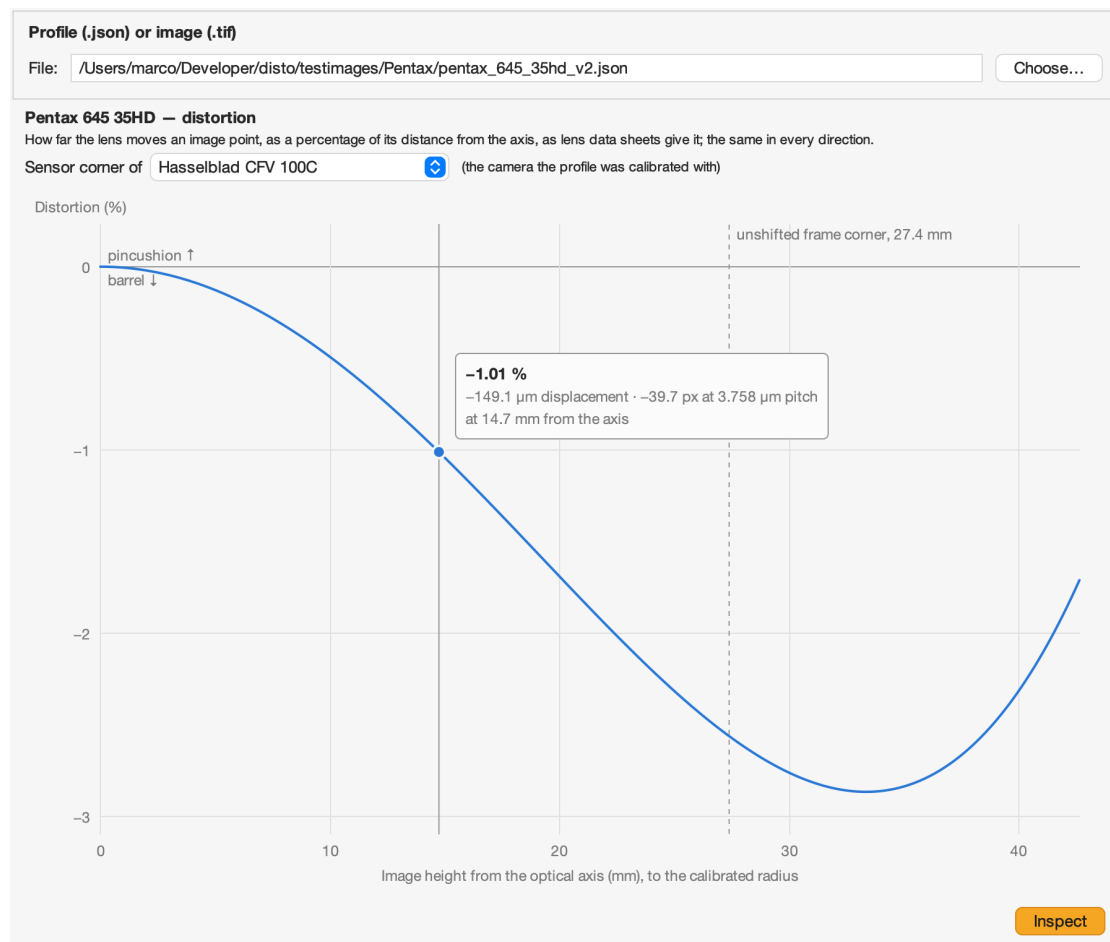


Figure 4: The Inspect tab showing a profile’s distortion, with the pointer resting on the curve.

4.6.1 A profile’s distortion chart

As soon as you choose a profile, the tab draws its distortion (Figure 4) the way lens data sheets do:

- The horizontal axis is the *image height*: the distance from the optical axis in millimetres, from the centre of the lens out to the radius the profile was calibrated to.
- The vertical axis is the *relative distortion*: how far the lens moves a point, as a percentage of that point’s distance from the axis. Below zero is barrel distortion (points pulled inwards); above zero is pincushion (points pushed outwards).

If the lens maker publishes a distortion curve, set it beside this one: the two should have the same general shape. Rest the pointer on the curve to read the exact value at that height, together with how far that moves a point on the sensor, in micrometres and in pixels at the calibration’s pixel pitch.

The unshifted frame. A dashed vertical line marks where the corner of an unshifted frame would fall: half the sensor’s diagonal. Without shift, the sensor only ever sees the part of the curve to the left of this line; everything to its right is reached only by shifting. The **Sensor corner** of list above the chart chooses which camera’s sensor to draw. It starts on the camera the profile was calibrated with or, if the profile does not name one, on a camera whose pixel pitch matches

the profile's; the note beside the list tells you which. Pick any other camera to see where its corner would fall with the same lens, or **None** to hide the line.

Decentred lenses. A profile calibrated with **p1**, **p2** describes distortion that differs from one direction to another, which a single curve cannot show. Such a profile is drawn instead as a surface over the whole image circle. Height and colour both show the relative distortion, blue for barrel and red for pincushion, with the scale beside it; the height is exaggerated so that the shape can be seen. Drag to turn and tilt the surface, and double-click to return to the starting view. Rest the pointer on the surface to read the distortion at that point, with its radial (towards or away from the axis) and tangential (sideways) displacements. The corner of the unshifted frame is drawn as a dashed ring on the surface.

4.6.2 Checking an image and a shift

Choose a TIFF, enter its shift (with the rear-standard boxes set as on the other tabs) and press **Inspect**. The console reports:

- the camera and pixel pitch disto found, and the export scale (whether the file is full size or downscaled);
- the orientation it used;
- where the optical axis falls, in pixels, and whether it lies outside the frame;
- how far the frame's corners reach from the axis; and
- which metadata will be carried through to the corrected file.

The optical axis position is the key line for checking a shift entry and the sign convention. A simple, reliable test uses vignetting, which is always centred on the optical axis. Take a frame of an evenly lit, plain surface with the lens stopped down (an LCC frame, for instance), inspect it with its shift, and compare: the reported axis position should match the centre of the vignetting, the point the darkening is centred on. If the axis is reported on the opposite side, a sign or a rear-standard box is wrong.

4.7 The Correct tab

In short: what this tab does

The Correct tab is the everyday part of disto. It takes the TIFFs your RAW developer exported, removes the lens's distortion from each one, and saves the results as new files in a folder of your choice.

You give it three things: the lens's profile (made on the Calibrate tab), the files, and the shift each file was taken at. For each file, disto works out where the optical axis fell on that frame, using the shift, and straightens the picture around that point, so straight lines in the scene come out straight in the picture.

Your original files are never changed. Each corrected copy keeps the original's colour profile and metadata, and records what disto did to it.

The form is filled in from top to bottom.

Profile. Choose the lens profile (**.json**), or drop it onto the field. Make sure it is the profile of the lens the pictures were taken with, calibrated at a similar focus distance.

Profile

Lens profile:

Shift readings

From the rear standard: ☐ Lateral (x) ☒ Rise/fall (y)

Images — shift of each shot; sx > 0 looks right, sy < 0 looks up

File	sx (mm)	sy (mm)
B0010950_X00_Y00.tif	0	0
B0010949_Xp18_Y00.tif	18	0
B0010953_X00_Ym18.tif	0	-18

Shift sx sy mm

Geometry

Orientation:

Pixel pitch (mm):

Sensor width (mm):

Output

Folder:

Framing:

Resampling:

Threads:

☐ Deflate-compress the output
☐ Continue despite a cropped-looking export
☒ Verbose report

Figure 5: The Correct tab, set up for three frames at different shifts.

Images. List the TIFFs and their shifts (§4.2). The shifts can all be the same, or differ from file to file. Two files with the same name, from different folders, cannot be corrected in one run, since their corrected copies would have the same name too; correct them in separate runs.

Geometry. Usually nothing to do here; see §4.3.

Output.

Destination folder Where the corrected files go. The folder is created if it does not exist yet. Each corrected file is named after its original with `_corr` added, so `B0010950.tif` becomes `B0010950_corr.tif`. **Open folder** shows the folder in the Finder.

Framing What to do at the edges of the picture. Straightening a picture moves its content: correcting barrel distortion, for instance, pushes the edges outwards, while correcting pincushion distortion pulls them in. So the corrected picture no longer fits the original rectangle exactly: some content falls outside it, and some areas inside it have no content. The three choices deal with this differently:

keep	The default. The output has the same size as the input, and the same scale. Content pushed outside the frame is cut off; areas left without content are filled with black.
fit	The output is cropped to the largest rectangle, with the original's proportions, that contains no black fill, and that crop is enlarged back to the input's size.
full	The canvas is enlarged so that the whole corrected picture fits, with black fill wherever there is no content. Nothing is cut off.

Tip. If you plan to stitch shifted frames (§5.3), use **keep** or **full**. **fit** enlarges each frame by its own amount, which depends on its shift, so the frames would no longer fit together by moving them alone.

Resampling How pixel values are calculated at their new positions. **lanczos3**, the default, keeps the most detail; **bicubic** is a little softer, and a little faster.

Threads How many processor cores to use. 0, the default, uses all of them; choose a smaller number to keep the Mac responsive for other work during a long batch.

Deflate-compress the output (slower) Off by default. The compression is lossless, but on 16-bit photographs it saves only a few percent of disk space while taking about four times as long. Six 100 MP frames came to 3.46 GB instead of 3.67 GB, in 78 seconds instead of 16.

Continue despite a cropped-looking export Corrects a file even though its proportions do not match the sensor's (§3.1). Use it only if you know the file is not cropped: a cropped file puts the optical axis in the wrong place, and the correction with it.

Verbose report Reports, for each file, the camera and pixel pitch used and where they came from, the export scale, the shift, the position of the optical axis in pixels, how far the corners reach from the axis, and how long the file took.

When corrected files already exist. If some of the corrected files are already in the output folder, from an earlier run for instance, *disto* asks what to do before it starts. The dialog lists those files and offers three choices: **Overwrite** replaces them with new corrections; **Skip**, the default, leaves them as they are and corrects only the remaining files; and **Cancel** stops without doing anything.

Safety. disto never overwrites one of your original files, whatever you choose. Each corrected file records, in its TIFF description, the disto version, the lens, the shift and the coefficients used, so you can always tell later how it was made. It also keeps the original's ICC colour profile and its EXIF and XMP metadata.

Warnings. Some situations do not stop the run, but deserve your attention. disto reports them in the console, and the status bar shows how many there were:

- **A frame reaches further from the optical axis than the profile was calibrated to.** The correction in the outermost corners is then extrapolated rather than measured; look at those corners carefully. A calibration with larger shifts would cover them.
- **A file was taken with a different camera from the one the profile was calibrated with** (reported once per camera). This is allowed, because the profile describes the lens alone, but check that the pixel pitch is right for that camera.
- **The input is 8-bit**, which can show banding in smooth gradients after resampling.

4.8 The Cameras tab

In short: what this tab does

The Cameras tab is disto's list of camera sensors. disto needs it because profiles measure in millimetres and images in pixels: to convert from one to the other, disto has to know the size of a pixel on the sensor (§3.3). It finds that by matching the camera name written in each TIFF against this list.

You will rarely need this tab. The list comes with common medium-format backs, and when you add a file from a camera that is missing, disto offers to add it for you (§4.4). Come here to look up an entry, correct a mistake, remove a camera, or add one by hand.

Cameras — matched to each TIFF by its EXIF make and model

Make	Model	Native (px)	Sensor (mm)	Pitch (µm)
Hasselblad	CFV 100C	11656 × 8742	43.8 × 32.9	3.758
Hasselblad	CFV II 50C	8272 × 6200	43.8 × 32.9	5.295
Hasselblad	X2D 100C	11656 × 8742	43.8 × 32.9	3.758
Phase One	IQ4 150MP	14204 × 10652	53.4 × 40.0	3.760
Phase One	IQ3 100MP	11608 × 8708	53.7 × 40.4	4.626
Phase One	IQ4 100MP Trichromatic	11608 × 8708	53.4 × 40.0	4.600
FUJIFILM	GFX100 II	11648 × 8736	43.8 × 32.9	3.760
FUJIFILM	GFX100S	11648 × 8736	43.8 × 32.9	3.760
FUJIFILM	GFX 50S	8256 × 6192	43.8 × 32.9	5.305

+ - Factory list — changes will be saved in ~/.disto/cameras.json Reset to Factory...

Hasselblad CFV 100C

Make:

Model:

Native size: × px

Sensor size: × mm pixel pitch 3.758 µm

Notes:

Revert Save

Figure 6: The Cameras tab with the factory list.

Unlike the other tabs, the Cameras tab does not run a command and has no orange button. Changes are saved straight away: when you press **Save** or **Add**, or remove a camera.

- Select a camera in the list to show its details in the form below. Each camera has:
 - Make, Model** The names the camera writes into the EXIF tags of its files, for example Hasselblad and CFV 100C. This is how disto recognises a file's camera, so they must match what is in the files; differences in capitals and spacing do not matter.
 - Native size** The width and height, in pixels, of a full-resolution export from this camera.
 - Sensor size** The width and height of the sensor in millimetres, as the camera's specifications give them, with width and height along the same sides as the native size. The pixel pitch shown beside it is the sensor width divided by the native width; that is the value disto uses. As a check, the width and the height must give the same pitch to within 1%; this catches a mistyped figure, or a width and height entered the wrong way round.
 - Notes** Anything you want to remember about the camera.
- To change a camera, edit its fields and press **Save**, or press **Revert** to throw your edits away. If you select another camera while there are unsaved edits, disto asks whether to save them.
- To add a camera, press +, fill in the form and press **Add**.
- To remove a camera, select it and press −; disto asks you to confirm first.
- **Reset to Factory...** replaces your list with the one disto ships with.

Where the list is kept. Until you first change anything, disto uses its built-in factory list, so cameras added in a new release of disto reach you automatically. Your first change saves the whole list to `~/.disto/cameras.json`, which from then on is your own list: you can edit and remove the factory entries just like your own, but new factory entries in later releases no longer appear by themselves. **Reset to Factory...** renames your file to `cameras.json.bak`, so nothing is lost, and goes back to the factory list. The command-line tool reads and changes the same file (§7.4).

5 Workflows

This section brings the tabs together into the two jobs you will actually do: calibrating a lens, and correcting a shoot. A third part explains how to stitch shifted frames once they are corrected.

5.1 Shooting a calibration series

Choose the scene. disto measures distortion by how much a straight edge bows. The bow is small, and it shows best along a *long* edge, so the ideal scene has broad boundaries between light and dark that run most of the way across the frame, in both directions: a building corner, bands of windows, a tiled or brick wall, a painted hoarding. Two kinds of subject work poorly:

- **Thin lines**, such as cables or the lines of a printed grid. Each has two edges a few pixels apart, which are hard to tell apart.
- **Fine repeating patterns**, such as a small checkerboard. Their edges are too short to bow measurably, so they carry almost no information.

A few long edges are worth more than many short ones.

Fix the focus. Set the focus to the distance you are calibrating for, usually infinity, and do not touch it during the series.

Shoot five to nine frames at different shifts. A good set is: one frame with no shift, four at full travel in each direction (left, right, rise, fall) and, if you can, four at the diagonals. The camera does not need to stay in the same place between frames, and the scene does not need to line up from one frame to the next: disto treats each frame's edges separately, and only the lens and its shift matter. Write down the shift of each frame as you go; naming the files after their shifts, as in `B0010949_Xp18_Y00.tif`, is a simple way to keep track.

The shifted frames are the heart of the calibration. A single unshifted frame is not enough, however good the scene, because all its edges sit within the same narrow range of distances from the axis, and the outer part of the image circle is never seen. Shifting moves the axis relative to the sensor, so each frame covers a different part of the image circle, and together they reach much further out.

Export the frames as described in §3.1, from the same RAW developer and with the same settings you will use for the pictures you later correct.

Calibrate on the Calibrate tab (§4.5). If the report flags a frame, check its shift entry first, then calibrate again.

Check the profile on the Inspect tab. If the lens maker publishes a distortion curve, the profile's curve should look like it. Then correct one of the calibration frames on the Correct tab and look closely at its edges: they should now be straight.

5.2 Correcting a shoot

1. Export the frames from your RAW developer as described in §3.1, with the same settings as the calibration frames.
2. If you are not sure of the rear-standard setting for your camera, check the shift entry of one frame on the Inspect tab first (§4.6.2). You only need to do this once per camera.
3. On the Correct tab, choose the profile, add the files with their shifts, tick the rear-standard boxes as needed, choose the output folder and press **Correct**.
4. When the run finishes, read the console for warnings, then open the folder and look at the results.

5.3 Stitching shifted frames

disto corrects each frame on its own; it does not join frames together. To combine shifted frames into one larger picture, stitch the corrected files afterwards in an image editor. How you stitch matters as much as the correction itself, because the wrong kind of stitch can bend the lines disto has just straightened.

Frames of one scene shot at different shifts are not like the frames of an ordinary panorama, where the camera turns to look in different directions. On a tech camera the camera does not turn: the lens or the back slides across one flat image plane. The frames are therefore pieces of one flat picture, which differ only by a sideways or up-and-down offset. Once disto has corrected them, they fit together by moving them alone, without any bending or warping. (With the back

shifted, the viewpoint stays exactly in place. With the lens shifted, the viewpoint moves by the amount of the shift, so objects close to the camera can move very slightly against distant ones; no stitch can remove that.)

Most panorama software assumes the opposite: it projects the frames onto a cylinder or a sphere and bends them to fit, which curves the straight lines *disto* has just made straight. As an example, six frames of a brick wall shot at shifts of up to 18 mm, stitched with such a projection, came out 628 pixels wider than the frames actually span, with gently curved mortar courses.

In Photoshop, use File → Automate → Photomerge:

- choose the Reposition layout, which aligns the frames by moving them only;
- untick Geometric Distortion Correction: the frames are already corrected;
- untick Vignette Removal, and deal with vignetting in your LCC workflow before exporting the frames instead;
- keep Blend Images Together ticked.

In other programs, such as Affinity Photo, look for an alignment that only moves the images, and avoid projections called cylindrical, spherical or perspective. Turn off any lens or distortion correction the program offers. If the program has no such option, you can place the corrected frames as layers and align them by hand: set the upper layer's blend mode to *Difference*, then move it until the overlapping area turns black, which means the two frames line up exactly. Then set the blend mode back to normal.

Why not place the frames using the shift values? The shifts tell *disto* where the optical axis lies, which is all the correction needs. But they are not precise enough to position frames against each other: 0.1 mm is 27 pixels on a 100 MP back, and a shift scale can be read to the nearest half millimetre at best. An image editor's alignment matches the frames' actual content instead, to a fraction of a pixel.

Check the result by following a long straight line across the joins. A doubled edge means two frames were not aligned; a curve means the program bent the frames.

Tip. For a large shoot, keep the shifts in a CSV file of `file,sx,sy` rows as you go, and load it with `Import shifts CSV...`. The same file works with the command line's `--shifts`.

6 Troubleshooting

The messages below are the ones you are most likely to meet, with what they mean and what to do about them.

“Cannot determine the pixel pitch” *disto* cannot convert between millimetres and pixels for this file, because its camera is not in the camera list, or the file carries no camera tags at all. Add the camera (§4.8), or give a pixel pitch or sensor width for the run (§3.3).

“The export looks cropped” The file's proportions do not match the sensor's, which usually means it was cropped in the RAW developer. Export the whole frame again, without cropping. If you are sure the file is not cropped, check that the camera entry's native size is right, then use `Continue` despite a cropped-looking export (`--force`).

“Only n straight edges were traced” The scene does not hold enough long straight edges for a dependable calibration. Choose a scene as described in §5.1.

“The longest straight edge found spans only...” The edges are too short to show how they bow. Photograph broad boundaries that run most of the way across the frame (§5.1).

“The fit did not converge” `disto` could not find coefficients that straighten the edges consistently. Check that the scene really has long straight edges, and that the shifts are right, including the rear-standard boxes.

A frame is flagged in the calibration report Its shift was probably entered wrongly; check it against your notes, correct it and calibrate again.

“...already exists” On the command line, `disto` does not replace an existing output unless you ask it to. Remove the file, choose another output, or add `--overwrite`. The app asks you instead (§4.7).

“...larger than 2 GB” Very large exports (BigTIFF), such as a 16-bit 150 MP file, are not supported yet. Export at a smaller size or in 8 bits.

7 Command-line reference

The command-line tool runs the same code as the app, and everything the app does can be done from a terminal or a script. Every command accepts `-h` or `--help`, and `disto help <command>` prints the same usage.

```
| disto <command> [options]
| disto help <command>
| disto --version
```

Options that take a value accept it after a space or an equals sign: `--shift 0,15` and `--shift=0,15` mean the same. When the first value is negative, use the equals form, as in `--shift=-12,10`; otherwise the minus sign makes `-12,10` look like an option of its own. Shifts are in millimetres and follow the sign convention of §3.2; `--rear` works in every command that takes a shift.

7.1 calibrate

Builds a profile from frames taken at different shifts; it is the command-line form of the Calibrate tab (§4.5).

```
| disto calibrate --lens "Rodenstock HR Digaron-W 40" --frames frames.csv \
|   -o digaron40.json
| disto calibrate --lens "HR Digaron-W 40" -o digaron40.json \
|   f1.tif:0,0 f2.tif:0,15 f3.tif:15,0 f4.tif:-15,0 f5.tif:0,-15
```

There are two ways to tell `disto` the shift of each frame. You can write it after the file name, as `file:sx,sy` (the shift follows the last colon in the argument), as in the second example. Or you can list the frames in a CSV file and give it with `--frames`, as in the first:

```
| file,sx,sy
| frame1.tif,0,0
| frame2.tif,0,15
| frame3.tif,0,-15
| frame4.tif,15,0
| frame5.tif,-15,0
```

File names in the CSV are looked up relative to the folder the CSV is in.

Option	Description
<code>--lens <name></code>	Lens name recorded in the profile. Required.
<code>-o, --output <file></code>	Where to write the profile. Required.
<code>--frames <csv></code>	Frames as <code>file,sx,sy</code> rows, instead of <code>file:sx,sy</code> arguments.
<code>--rear <x y xy></code>	Shift readings on these axes come from the rear standard, and are inverted.
<code>--focal-length <mm></code>	Focal length, recorded for reference.
<code>--focus <inf metres></code>	Focus distance the profile is calibrated at. Default <code>inf</code> .
<code>--developer <name></code>	RAW developer used for the exports, recorded in the profile.
<code>--k3</code>	Also fit the sixth-order radial term k_3 .
<code>--tangential</code>	Also fit the decentring terms p_1 and p_2 .
<code>--no-k2</code>	Fit only k_1 , for a lens with very simple distortion.
<code>--pixel-pitch <mm></code>	Pitch of the files' pixels, when the camera is not in the list: the sensor's pitch for a full-resolution export, twice it for a half-size one (§3.3).
<code>--sensor-width <mm></code>	Sensor width; the pitch is this over each file's width in pixels, so only for full-resolution exports.
<code>--orientation</code> <code><auto 0 90 180 270></code>	Sensor rotation. Default <code>auto</code> , from EXIF.
<code>--force</code>	Proceed despite too few frames, too few or too short edges, or a fit that did not converge.
<code>--verbose</code>	Also report each frame's axis position and the residual by radius.

All frames of one calibration must come from one camera and be exported at one size; `disto` refuses frames that resolve to different pixel pitches.

7.2 correct

Applies a profile to exported TIFFs, given the shift of each shot; it is the command-line form of the Correct tab (§4.7).

```
| disto correct -p digaron40.json --shift 0,15 -o out.tif in.tif
| disto correct -p digaron40.json --shifts shifts.csv -o corrected/ *.tif
```

The meaning of `-o` depends on the inputs. With one input, it names the output file, as in the first example. With several inputs, or when it names a folder that already exists, it is an output folder (created if needed), and each output is named `<name>_corr.tif`.

The shifts can be given in two ways. `--shift` gives one shift for every input. `--shifts` gives a shift per file, in a CSV of `file,sx,sy` rows. The rows are matched to the inputs by file name alone, so the CSV can sit in any folder:

```
| file,sx,sy
| _DSC1001.tif,0,0
| _DSC1002.tif,0,15
| _DSC1003.tif,-12,10
```

Option	Description
<code>-p, --profile <file></code>	Lens profile to apply. Required.
<code>-o, --output <file folder></code>	Output file, or a folder for several inputs. Required.
<code>--shift <sx>,<sy></code>	Shift in mm for every input.
<code>--shifts <csv></code>	Per-file shifts as <code>file,sx,sy</code> rows. Give this or <code>--shift</code> .
<code>--rear <x y xy></code>	Shift readings on these axes come from the rear standard, and are inverted.
<code>--frame <keep fit full></code>	Output framing (§4.7). Default <code>keep</code> .
<code>--filter <lanczos3 bicubic></code>	Resampling kernel. Default <code>lanczos3</code> .
<code>--pixel-pitch <mm></code>	Pitch of the files' pixels, when the camera is not in the list: the sensor's pitch for a full-resolution export, twice it for a half-size one (§3.3).
<code>--sensor-width <mm></code>	Sensor width; the pitch is this over each file's width in pixels, so only for full-resolution exports.
<code>--orientation</code> <code><auto 0 90 180 270></code>	Sensor rotation. Default <code>auto</code> , from EXIF.
<code>--threads <n></code>	Worker threads; 0, the default, uses every core.
<code>--compress</code>	Write Deflate-compressed TIFFs.
<code>--overwrite</code>	Replace outputs that already exist, instead of refusing them. Inputs are never overwritten.
<code>--force</code>	Continue despite an export that looks cropped.
<code>--verbose</code>	Report the camera, pitch, axis position and timing for each file.

7.3 inspect

Describes a profile, or the geometry disto works out for an image at a given shift; it is the command-line form of the Inspect tab (§4.6), without the chart. A `.json` file is read as a profile, anything else as a TIFF.

```
disto inspect digaron40.json
disto inspect in.tif --shift 0,15
disto inspect in.tif --rear y --shift 0,-18
```

Option	Description
<code>--shift <sx>,<sy></code>	Shift to place the optical axis for. Default 0,0.
<code>--rear <x y xy></code>	Shift readings on these axes come from the rear standard, and are inverted.
<code>--pixel-pitch <mm></code>	Override the pixel pitch: the pitch of this file's pixels.
<code>--sensor-width <mm></code>	Sensor width; the pitch is this over the file's width in pixels, so only for full-resolution exports.
<code>--orientation</code> <code><auto 0 90 180 270></code>	Sensor rotation. Default <code>auto</code> , from EXIF.

7.4 cameras

Lists the cameras disto knows, or changes the list; it is the command-line form of the Cameras tab (§4.8) and reads and writes the same file, `~/.disto/cameras.json`.

```
disto cameras                # list the cameras in effect
disto cameras --add my-camera.json  # add or replace entries
disto cameras --remove "CFV II 50C"  # remove an entry
disto cameras --reset            # back to the factory list
```

Option	Description
<code>--add <file></code>	Add the entries in this JSON file, replacing any with the same make and model.
<code>--remove <camera></code>	Remove a camera, named by model or by make and model. Repeatable.
<code>--reset</code>	Go back to the factory list; yours is kept as <code>cameras.json.bak</code> .

The file given to `--add` holds a list of cameras:

```
[
  {
    "make": "Hasselblad",
    "model": "CFV 100C",
    "native_size_px": [11656, 8742],
    "sensor_size_mm": [43.8, 32.9],
    "notes": "optional"
  }
]
```

Each entry needs `model`, `native_size_px` as [width, height], and `sensor_size_mm` as [width, height] in millimetres along the same sides. An entry may give `pixel_pitch_mm` instead of the sensor size, as files from earlier versions do. `make` and `notes` are optional. `--add` also accepts a whole `cameras.json` copied from another machine.

7.5 The profile file

A profile is a JSON file. You never need to edit it, but it is easy to read:

```
{
  "schema": 2,
  "lens": "Rodenstock HR Digaron-W 40mm",
  "focal_length_mm": 40,
  "focus_distance_m": null,
  "calibrated_ic_radius_mm": 42.5,
  "model": "brown-conrady",
  "units": "mm",
  "radial": { "k1": -1.2e-5, "k2": 3.4e-9, "k3": 0.0 },
  "tangential": { "p1": 0.0, "p2": 0.0 },
  "calibration": {
    "camera": "Hasselblad CFV 100C", "pixel_pitch_mm": 0.00376,
    "frames": 9, "lines": 143, "rms_px": 0.31, ...
  },
  "tool_version": "0.1.0"
}
```

`focus_distance_m` is null for infinity, and `calibrated_ic_radius_mm` is how far from the axis the calibration frames reached (the *calibrated to* figure of the report). The coefficients describe the Brown–Conrady model in millimetres about the optical axis: an ideal point at (x, y) , with $r^2 = x^2 + y^2$, is recorded by the lens at

$$\begin{aligned}
x_d &= x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + p_1(r^2 + 2x^2) + 2p_2 xy, \\
y_d &= y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + 2p_1 xy + p_2(r^2 + 2y^2).
\end{aligned}$$

7.6 Exit codes

Code	Meaning
0	Success
1	Usage error
2	Input or precondition failure
3	The fit did not converge (calibrate)