Version 2026.09.0

Digital image correlation, measured carefully

What a DIC measurement actually is, what it can and cannot tell you, and how to run one that holds up. The manual for SurView DIC.

SurView DIC · version 2026.09.0 · 2026-09-03
A Katalyst Nord project. Free and open source, LGPL-2.1-or-later.
github.com/katalystnord/SurView

Contents

1. Two photographs and a question

Digital image correlation answers one question: how did the surface in this photograph move, compared with this other one? Everything else in this manual is about what that question does and does not let you ask.

The reference and the target

Every measurement starts with two photographs of the same surface: a **reference**, taken before whatever you are studying happens, and a **target**, taken after. A test with several load steps has several targets, each compared against the reference in turn - Part II covers what "compared" means once there is more than one.

The surface has to carry a random, high-contrast texture: a **speckle pattern**. This is not decoration. It is what makes the measurement possible at all. A patch of featureless grey has nothing distinguishing it from the patch next to it, so there is no way to say which pixel in the target corresponds to which pixel in the reference - the correspondence is genuinely ambiguous, not merely hard to compute. A patch of speckle is different every few pixels in every direction, so its match in the target is (with enough contrast and enough texture) unique. Part V returns to what makes a pattern good.

The subset, and the grid

DIC does not try to match single pixels. It lays a grid of points over the reference image, and around each point takes a small square patch of pixels called a **subset**. For each subset, it searches the target image for the patch of pixels whose pattern of light and dark most closely matches it - allowing that patch to have moved, stretched, rotated, or sheared slightly, since the surface genuinely did those things between the two photographs. The centre of the best match is where that point of the surface now is. The distance between where it started and where it was found is the **displacement** at that point, in pixels.

Do this at every point of the grid and the result is a **displacement field**: not one number, but a value at every measured point, which can be drawn as a picture - usually a colour map, warm where the movement was large, cool where it was small.

How sure the match is

The search does not just report a best match; it reports how good that match was, as a **correlation coefficient**. A value close to its best possible score means the patch in the target looks almost identical to the patch in the reference, once the found deformation is accounted for. A poor score means no candidate patch looked convincingly like the original - the surface may have torn, gone out of focus, left the frame, or simply have too little texture in that subset to pin down at all.

This number is a statement about pattern-matching confidence, not a statement about physical truth. A subset can match with high confidence while telling you nothing you actually want to know - a smooth, empty patch of background can correlate excellently against itself while genuinely reporting no useful information. Confidence in the match and meaningfulness of the answer are related but not the same question, and Part III is built entirely around keeping them separate.

> **IN ONE SENTENCE**
>
> DIC follows small, uniquely textured patches of a photographed surface from a reference image to a target image, and reports how far each one moved and how confident it is in that answer.

Two-dimensional, stereo, and volumetric

"DIC" names a family of techniques rather than one, and they differ in what dimensions of movement they can see at all. The distinction matters before anything else in this manual, because it decides which questions a given measurement is even capable of answering.

Two-dimensional DIC: one camera, movement in the image plane

One camera looks squarely at a flat, or nearly flat, surface, and measures how that surface moves *parallel to the image plane*: two components of displacement, in-plane, at every point. This is what everything else in this manual describes, and what SurView measures today.

Its limitation is exactly its definition: it cannot see movement *toward or away from* the camera. Worse, it cannot tell such movement apart from real in-plane strain. A specimen that bows slightly out of plane appears, to a single camera, to have stretched - because the part of it that moved closer genuinely does image larger. That is not an implementation flaw to be fixed; it is information a single viewpoint does not contain. It is why a 2D measurement wants a specimen that stays flat, and why out-of-plane motion belongs on the list of things a noise floor cannot warn you about (Chapter 9).

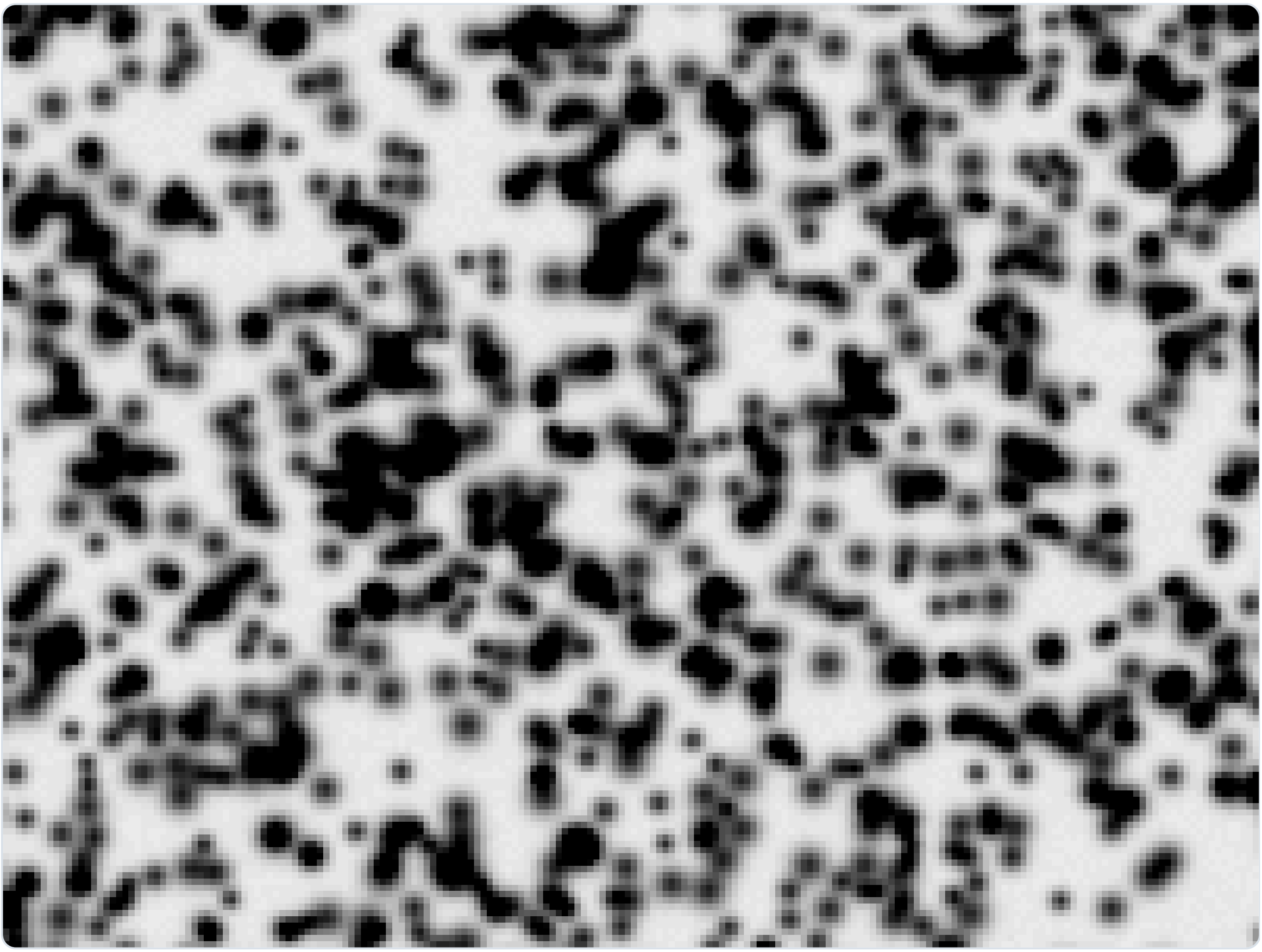
Stereo DIC: two cameras, movement in three directions

Two calibrated cameras viewing the same speckled surface from different angles can triangulate each point, and so measure all three components of its displacement, out-of-plane included. This is universally called "3D DIC," and the name slightly overstates it: what it measures is a **surface**, one depth per point of the master camera's view, carrying a fully three-dimensional displacement. The geometry is two-and-a-half dimensional; the displacement on it is genuinely 3D. It cannot see inside the material, only the face turned toward the cameras.

Volumetric DVC: correlation through the inside of a solid

Digital *volume* correlation applies the same idea to a three-dimensional image of a specimen's interior, usually from X-ray computed tomography: instead of subsets of pixels on a surface, it follows subsets of *voxels* through the volume, and reports displacement and strain inside the material rather than on its skin. Natural texture in the material's own microstructure - porosity, inclusions, grain contrast - takes the place of an applied speckle pattern, since nothing can be painted onto an interior.

> **WHICH ONE THIS MANUAL COVERS**
>
> **Two-dimensional DIC only.** That is what SurView measures today, and the rest of this manual is written for it throughout. Stereo and volumetric correlation are both real, tracked work on the roadmap - the correlation engine underneath SurView already carries much of what each needs - but neither has a workflow in the application yet, and a chapter describing a screen that does not exist would be worse than an honest gap. Read anything here about displacement, strain, or reliability as in-plane unless it says otherwise.



A speckle pattern, closely cropped. Random and non-periodic, with strong local contrast at every scale a subset might span. Every small window onto this pattern is different from every other one, which is what lets a correlation search find exactly one convincing match rather than several equally good ones.

2. Displacement is measured, strain is fitted

Displacement and strain are reported side by side and look like the same kind of number. They are not measured the same way, and treating them as though they were is the single most common way to misread a DIC result.

Displacement exists at a point

A correlation finds where one subset went. That answer belongs entirely to that one point: ask "how far did this point move" and the field has a direct answer, measured by comparing that point's own neighbourhood in the reference against the target.

Strain does not

Strain is a measure of local stretching or shearing - how much a small patch of material deformed relative to its own neighbours, not how far it travelled. That is a *gradient* of displacement, and a gradient has no meaning at a single point on its own: it is a comparison between a point and the points around it. So strain is not measured directly at all. It is **fitted**: a plane (or a more general surface) is fitted through the displacements of every measured point inside a small neighbourhood, and the slope of that fit is the strain reported at the centre.

This has real consequences, and each of them is easy to miss unless a tool goes out of its way to surface it.

A declined fit is not a strain of zero

A fit can fail to find enough well-correlated neighbours to fit through at all. When that happens, the honest answer is "not established here", and a tool that quietly reports zero in that case is lying in the most dangerous possible direction: zero looks like a real, specific, reassuring answer - "this region did not strain" - rather than "this region was never evaluated". SurView carries the distinction explicitly, as a flag alongside every strain value, and a strain map with holes in it means exactly what it looks like: places nothing was established, not places nothing happened.

Neighbourhoods can be silently substituted

A fit needs a minimum number of neighbouring points inside its subregion. If a subregion does not contain enough of them - near an edge of the measured area, or near a hole, or wherever the displacement field itself is sparse - some engines will quietly reach further out and borrow the nearest points available instead of refusing, and the result looks exactly as complete as a properly supported fit. The honest response is to say, live, how many points a given subregion setting will actually gather before a run is even started, so a setting that would trigger this silently is visible before it is used rather than discovered afterwards.

Only well-correlated points get to vote

A point that correlated poorly is not trustworthy evidence about its own displacement, and a strain fit that used it anyway would be building a gradient on top of noise. DIC engines typically exclude points below a correlation threshold (0.9 is a common default) from every fit they could otherwise contribute to. A consequence worth expecting rather than being alarmed by: a strain map is often visibly sparser than the displacement map sitting right next to it. That is the fit being conservative, not a fault.

The trap that is easiest to fall into: extrapolation dressed as measurement

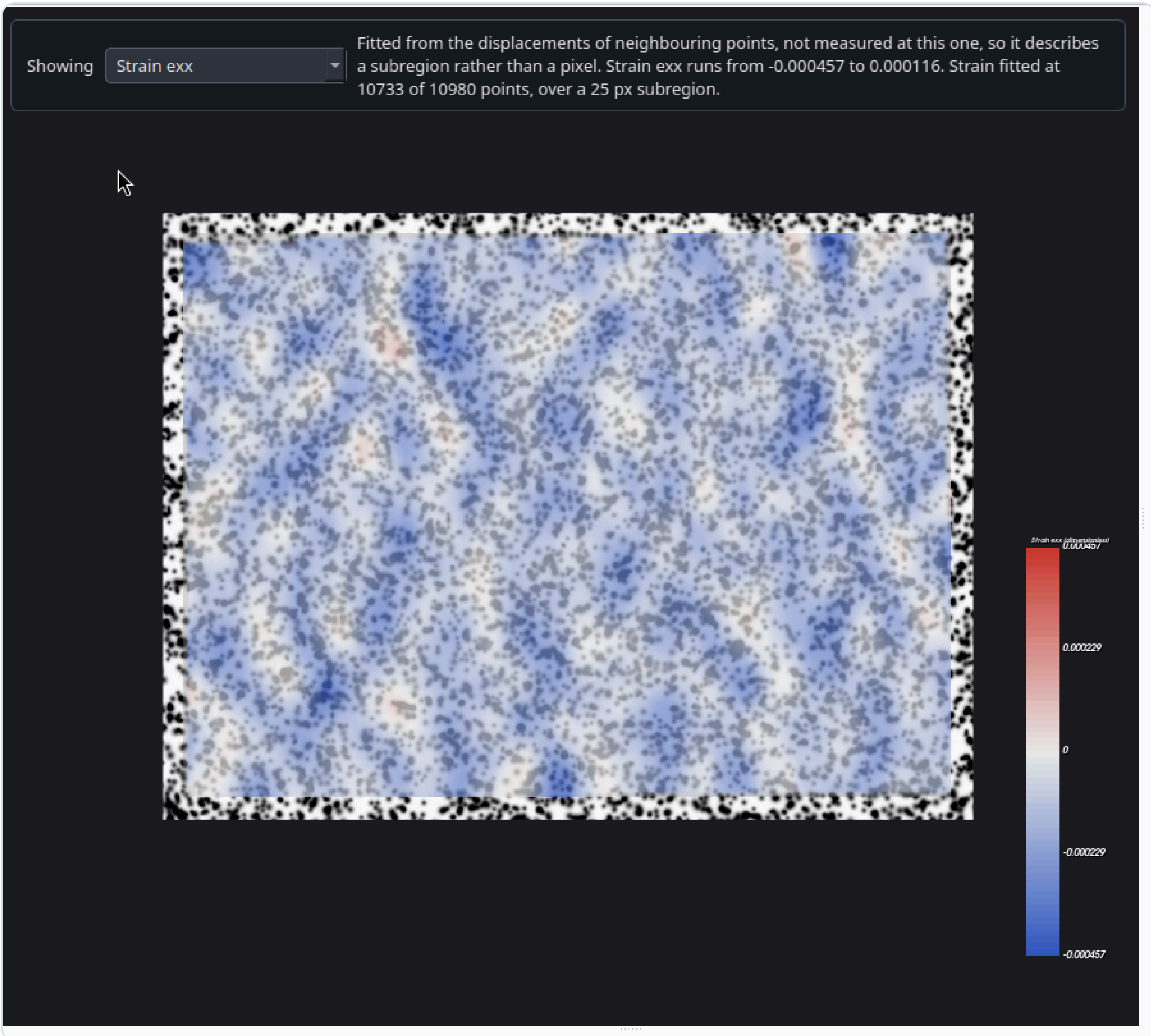
Here is the one worth reading twice. A strain fit is centred on a point and built from that point's *neighbours*. Nothing stops the fit from succeeding at a centre point whose own displacement was *rejected*: the centre simply excludes itself from its own regression and still receives a strain value extrapolated entirely from points around it. That value describes a place the instrument never actually got a reading from, and on the screen it is completely indistinguishable from a value measured where the instrument really did succeed.

This was found, not anticipated: a run reported **"strain fitted at 1092 of the 1025 solved points"** - more strain values than there were successful displacement measurements to fit them from, which is only possible if some of those strain values were sitting on points the instrument had already given up on. The fix is a rule that is nowhere in the correlation engine itself and has to be added on top: **strain is reported only at a point whose own displacement was measured.** A rejected centre gets no strain value, however good its neighbours look, because a value there would be an extrapolation wearing the same colour as a measurement.

> **WHAT TO CHECK, EVERY TIME**
>
> Before reading a strain map as an answer, check: are unmeasured regions shown as gaps, or could they be reading as zero? Does the strain map extend anywhere the displacement map itself has holes? If a tool cannot answer either question, treat every value near a boundary or a hole with real suspicion.

Two display choices that follow from this

Because strain and displacement are different kinds of quantity, they should not share a colour convention. A strain field's scale is centred on zero, because zero strain is a real physical state (nothing stretched) and the sign either side of it is meaningful (tension against compression). A displacement field's scale is not centred, because a displacement of zero is only "wherever the reference frame happens to sit" - a specimen that simply translated across the frame would spend half its colour range on values that cannot occur if the scale were forced to straddle zero.



A rotation is not a strain. This specimen was turned, not stretched, and the correct answer is zero strain everywhere. The fitted field reads -0.000457 to 0.000116 across the whole specimen: zero, to within the method's own noise. A shape function that mistook rotation for strain would read around 0.017 here, the sine of the angle, and it would look entirely plausible on a colour map.

3. A rejected point is not a zero

One principle, showing up in every part of a DIC result: a place the instrument did not measure, and a place the instrument measured a value of zero, must never look the same. This chapter states the principle once so the rest of the manual does not have to keep re-arguing it.

Why zero is the dangerous default

Software defaults numeric fields to zero constantly, and it is usually harmless. In a measurement field it is not, because zero is never a neutral placeholder here - it is always *also* a legitimate, meaningful answer. Zero displacement means "this point did not move." Zero strain means "this material did not deform." A rejected point defaulting to zero is therefore not a blank; it is a false and entirely plausible-looking claim, and it looks exactly as confident as a real one.

The fix is the same everywhere it applies: use a value that cannot be mistaken for a measurement - not-a-number, a missing entry, a hole in a colour map - and carry a separate flag saying whether a value was actually established. "Nothing here" and "zero here" have to stay two different, unconfusable states, all the way from the engine to the screen to whatever file the result is exported into.

Where this shows up

- **A rejected point's displacement.** The correlation failed, or the subset moved out of frame, or the match was too poor to trust. The point is marked as not converged, and reports no displacement value at all - not a displacement of zero.
- **A declined strain fit** (Chapter 2): the same rule, one derived quantity further out.
- **An unestablished reliability figure** (Part III): a noise floor or match-conditioning value that could not be computed is absent, never zero - zero would be the single most flattering, and therefore most misleading, reading either metric could give.
- **A frame a sequence could not read** (Part IV): missing from a curve as a visible break, never plotted as though the quantity had returned to zero.
- **An exported field** (Chapter 12): the same absence has to survive leaving the application. A viewer that assumes every cell holds a number, and a format that cannot represent "not established", quietly turn every one of the cases above back into a lie the moment the file is opened somewhere else.

A READING HABIT WORTH HAVING

When a field looks unusually smooth or unusually complete, ask what would have happened to a point the instrument could not measure. If the honest answer is "it would look identical to a real zero," the field is not to be trusted at face value, whichever tool produced it.

4. One coordinate frame

Every number in a DIC result is a position or a displacement, and every position or displacement is meaningless without knowing which way the axes point. This is duller than the rest of Part I and just as capable of quietly ruining a result.

Image coordinates, not maths coordinates

A photograph is stored as rows of pixels, and the row order is a convention, not a law: some formats store the top row of the image first, others store the bottom row first. Screen and image coordinate systems follow the file: x increases to the right, and **y increases downward**, with the origin at the top-left pixel. That is the opposite of the y-axis convention used in ordinary graphing, where y increases upward - and a rendering system built for graphs (as most 3D and plotting toolkits are) will get this backwards unless it is deliberately told not to.

Why this has to be nailed down explicitly, once

Different image formats do not even agree with each other. Some readers honour a file's own stated orientation; others always deliver rows in one fixed order regardless of what the file says. Left alone, this produces a genuinely alarming failure mode: some image formats display correctly and others display **vertically mirrored**, silently, depending only on which file format happened to be used for that particular photograph - with nothing about the mirrored image looking obviously wrong at a glance, since a flipped photograph of a speckle pattern still looks like a perfectly plausible speckle pattern.

The only reliable fix is to normalise every image, on the way in, to one stated row order, and to make every other coordinate in the system (a click on screen, a region's corners, a measured point, a rendered field) agree with that same order and with nothing else. Once that is true, a click on the picture, a point in the result, and a pixel in the exported file are all talking about the same place without any conversion between them anywhere in the pipeline - which is also what makes it safe to overlay a result directly on top of its own reference photograph and trust that they line up.

Why an exported file has to say so explicitly

A result rarely stays inside the tool that produced it. It gets opened in a general-purpose viewer that has its own, entirely reasonable, default assumption about which way is up - usually the graphing convention, not the image convention. If the file does not state its own coordinate frame, a viewer that assumes the wrong one draws the field **mirrored**, and there is no way to tell from looking, because a mirrored field of a real specimen still looks like a plausible field of some specimen. The one thing that catches it is laying the exported field directly back over the original photograph: a field that lines up is right, and a field that lines up perfectly except flipped is a coordinate-frame bug wearing the appearance of a correct result.

5. The region, and what a hole means

Most of the time a correlation does not need to measure the whole photograph - the specimen fills only part of the frame, and the background around it correlates against itself perfectly well while telling you nothing you want to know. Restricting where points are placed is not just an optimisation; it changes what the result can honestly claim.

Drawn or detected, the region is honest about which

A region of interest can be drawn by hand, corner by corner, or proposed automatically by segmenting the image for wherever the speckle is strong enough to correlate well. The two are not the same kind of claim: a hand-drawn boundary is a person's judgement about where the specimen is, and a detected one is an algorithm's judgement about where the pattern is measurable, which are related but different questions. A result should carry which kind of boundary produced it, because a reader deciding how much to trust the edge of a field needs to know which question was actually asked there.

What a region actually restricts

A region decides where the *centres* of measurement points are allowed to sit. By itself it does not shrink the subset each point correlates over: a point near the edge of a region still looks at a full subset of pixels around it, some of which lie outside the region. So a boundary is softer than it looks, and points close to it are influenced by whatever sits just beyond it.

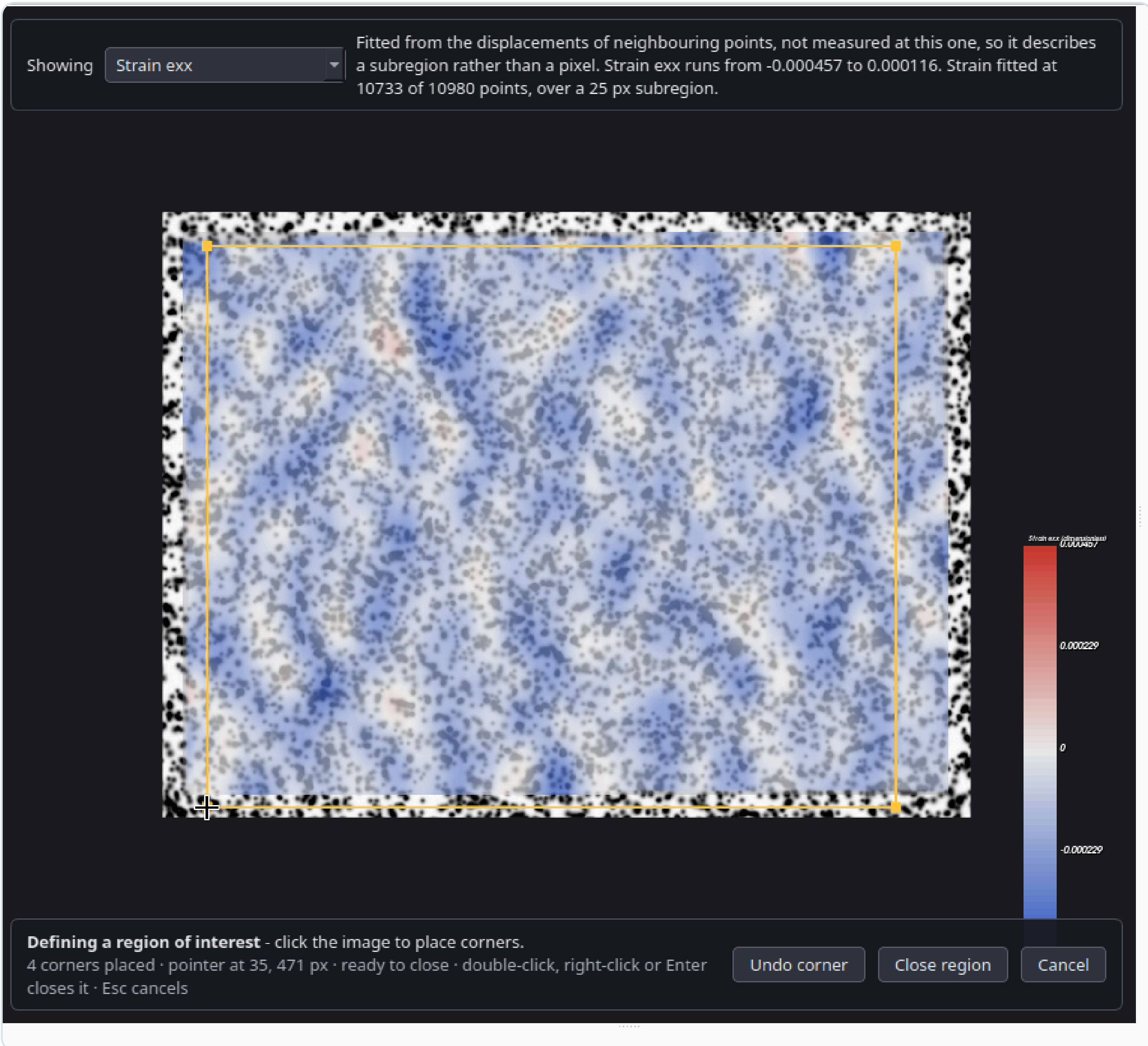
That softness is not inevitable, though it is what most tools do. A correlation can instead be run on only the *valid* pixels of a subset that straddles a boundary, masking out the rest, at the cost of having fewer pixels to work with at exactly the places where the measurement is already hardest. Whether that trade is worth making depends on what lies outside the boundary: more specimen you simply chose not to measure is fairly harmless to include, and a hole showing the background behind the specimen is not. Ask of any tool which of the two it does, because a full subset near a boundary and a masked one produce different numbers there and neither is labelled.

A hole is not decoration

Many real specimens are not solid: a dogbone tension coupon drilled for a pin, a plate with a hole for a fastener, a bracket with slots cut into it. Where the material is genuinely absent, the camera sees whatever is behind the specimen instead - background, or a fixture, or empty space - and that background does **not** move the way the specimen does. A measurement point placed there correlates the background against itself perfectly well and reports, with complete confidence, that nothing moved.

That is the worst possible place for a false reading to appear. A hole through a loaded part is exactly where stress concentrates, which is exactly where a strain map is most worth looking at closely - and a cold, confident, entirely wrong "no strain here" sitting right at that concentration is precisely the kind of error nobody thinks to double-check, because it looks like an answer rather than like a mistake.

A region that can describe a hole - an outer boundary with a place cut out of it, excluded from measurement - fixes this by simply not placing points there at all. What is drawn is then a positive statement about the specimen's actual shape, not merely a rough outer boundary hoping nobody looks too closely at what is inside it.



Drawing a region. Corners are placed by clicking the image, and the mode states what it is doing, how many corners exist so far, and every way to finish or abandon it. A hole is added the same way, as a second ring cut out of the region already in force.

A subset can still reach into a hole it does not sit inside

The same softness that applies at the outer boundary applies at a hole's edge too, and for the same reason: a subset extends beyond the single point it is centred on. A point just outside a hole can still have a subset that reaches partway into it, picking up some pixels that are genuinely background and do not move with the specimen. That pulls its answer very slightly toward "did not move," in the same quiet, plausible-looking direction as a point placed fully inside the hole would be, just to a lesser degree.

The honest response is not to pretend this cannot happen by excluding every point anywhere near a hole - that would apply a stricter rule at a hole's edge than the same result already accepts at the outer boundary, for no principled reason. It is to **count and report** how many measured points have a subset reaching into a hole, so a reader can widen the exclusion or simply read those points knowing what they are.

6. Sequences: order matters, reference matters

A DIC test is almost never one photograph compared against another. It is a loading series: a reference, and a run of targets taken as whatever is being studied happens over time. Two decisions that look like bookkeeping turn out to be about correctness.

Frame order is a correctness problem, not a presentation one

Sort a list of filenames the way a computer sorts text, and `frame_10` comes before `frame_2`, because `"1"` is a smaller character than `"2"` and the rest of the comparison never looks at the full number. A twelve-frame test sorted that way gets measured in the order 1, 10, 11, 12, 2, 3, and so on - and every individual frame still solves correctly, because the correlation itself has no idea what order it was asked to run in.

The result is a field that is internally perfect and, taken as a series, complete nonsense: a specimen that appears to jump backward and forward instead of loading smoothly. Nothing about any one frame reveals the problem, because nothing is wrong with any one frame. The only fix is to sort frames the way a person reads them - as numbers, not as text - before anything is measured, so what is listed as the sequence is guaranteed to be what gets measured as the sequence.

The same trap appears again on the way out. A results file numbered without leading zeros sorts the same wrong way in most viewers, so an exported frame 10 is grouped before frame 2 when a series is played back as an animation. Files need to be numbered so their names sort the same way their content is meant to play.

Every frame against the original reference, by default

The straightforward, and usual, meaning of a DIC sequence is that every target is compared against the *same* reference frame, not against the frame immediately before it. That keeps every reported displacement directly comparable across the whole test: frame 5's displacement and frame 9's displacement both mean "how far has this point moved from where it started," and can be laid over one another or subtracted from one another meaningfully.

The cost is that correlation quality degrades as the specimen deforms further from how it looked in that original photograph - the later the frame, the less it resembles the reference, and eventually subsets that correlated beautifully at the start start failing. The next chapter is about what to do when that happens.

7. When the reference goes stale

A reference that no longer resembles the current state of the specimen makes correlation fail, not through any flaw in the method, but because the two images genuinely no longer look alike enough to match. Re-anchoring is the standard answer, and it trades one kind of correctness for another in a way worth understanding before switching it on.

What re-anchoring does

Once too little of the field still correlates well against the original reference, later frames are measured against the *current* frame instead of the original one. Each point's displacement since the original reference is banked - the new increment is added to whatever was already accumulated - so what is **reported** stays relative to the original reference throughout, even though what was actually **measured** on any individual frame after the switch was relative to a more recent one.

This recovers correlation on a specimen that has deformed too far to compare directly against its starting photograph. It is not free: it abandons any point that could not be measured on the exact frame the re-anchor happens on, because there is no other way to know where that point went. That trade is explained wherever the setting is switched on, not buried in a default.

The rule that decides when to re-anchor has to count losses, not just successes

The obvious way to decide "has the reference gone stale" is to ask what share of points still correlate well. The obvious way to build that is to count the points that succeeded. That is backwards, and the reason is not obvious until you watch it fail: if only the points that are *still solving well* get a vote, then a field that has lost half its points to decorrelation but still correlates beautifully on the half that remains never triggers a re-anchor at all - the surviving half looks great and is the only half being asked.

On a real synthetic tension sequence this was not a hypothetical: solved points fell from 97% of the field to 47% across five load steps, and a rule that counted only successes never re-anchored once. **A point lost to decorrelation is the strongest evidence available that the reference has gone stale**, and it has to count against the reference exactly because it dropped out, not despite it.

There is one case where that rule has to bend the other way: a frame where *nothing* correlated must never trigger a re-anchor, because re-anchoring works by banking the increment just measured, and a frame with no increment to bank marks every point lost and ends the sequence's usefulness outright. A reference that is merely stale can recover on a later frame. A field that has lost every tracked point cannot.

The threshold is set by measurement, and moving it changes the outcome directly

A convenient-sounding default (borrowed from other tools' 75% share) turned out to be too permissive when tried against real data: at 75%, a tension sequence sat at 75.3% on its fourth load step - a hair above the line - held the reference one step too long, and arrived at the fifth step with less than half the field still solved. At 90%, the same sequence re-anchors one step earlier, and its final frame keeps far more of the field intact. The number is not a convention; it is a knob that visibly changes how much of a hard test survives to the end.

Two rules the bookkeeping has to get right, always

- **A frame is always reported on the original grid, relative to the original reference** - whatever it was actually measured against. Otherwise the reported field drifts across the picture as the specimen itself does, and no two frames of the series, and no frame and the original photograph, can ever be laid over one another again.
- **A point with no measurement on a re-anchor frame is lost, not frozen.** Its position in the new reference is unknown the moment the anchor moves without it, so continuing to report its last known displacement forever would present a region that has genuinely stopped being tracked as though it had simply stopped moving - a measurement claim, sitting where there is none left to make.

8. Points that fail, and the pass that repairs them

Most points a correlation loses are not places where the surface is genuinely unmeasurable. They are places where the search started from a poor guess. A second pass that gives failed points a better starting guess, and re-solves them properly, recovers a substantial share of a field - if it is built so that it can never make anything worse.

Borrowing from the neighbours, then re-solving in full

A subset's correlation search has to start somewhere, and if that starting guess is far from the real answer, the search can fail to converge even though a perfectly good match exists nearby. A recovery pass fits a displacement field to the points around a failed one that *did* correlate well, uses that fit as a better starting guess at the failed point, and runs the full correlation search again from there.

The result of the fit is never reported as a measurement on its own. It is only ever an initial guess, and the point is only accepted once it has been correlated properly from that guess - with its own, independently earned correlation score. Reporting the fitted value directly, without re-solving, would put an interpolation borrowed from a point's neighbours into the field wearing the same appearance as a real measurement, which is exactly the trap Chapter 2 describes for strain, one step further along the same idea.

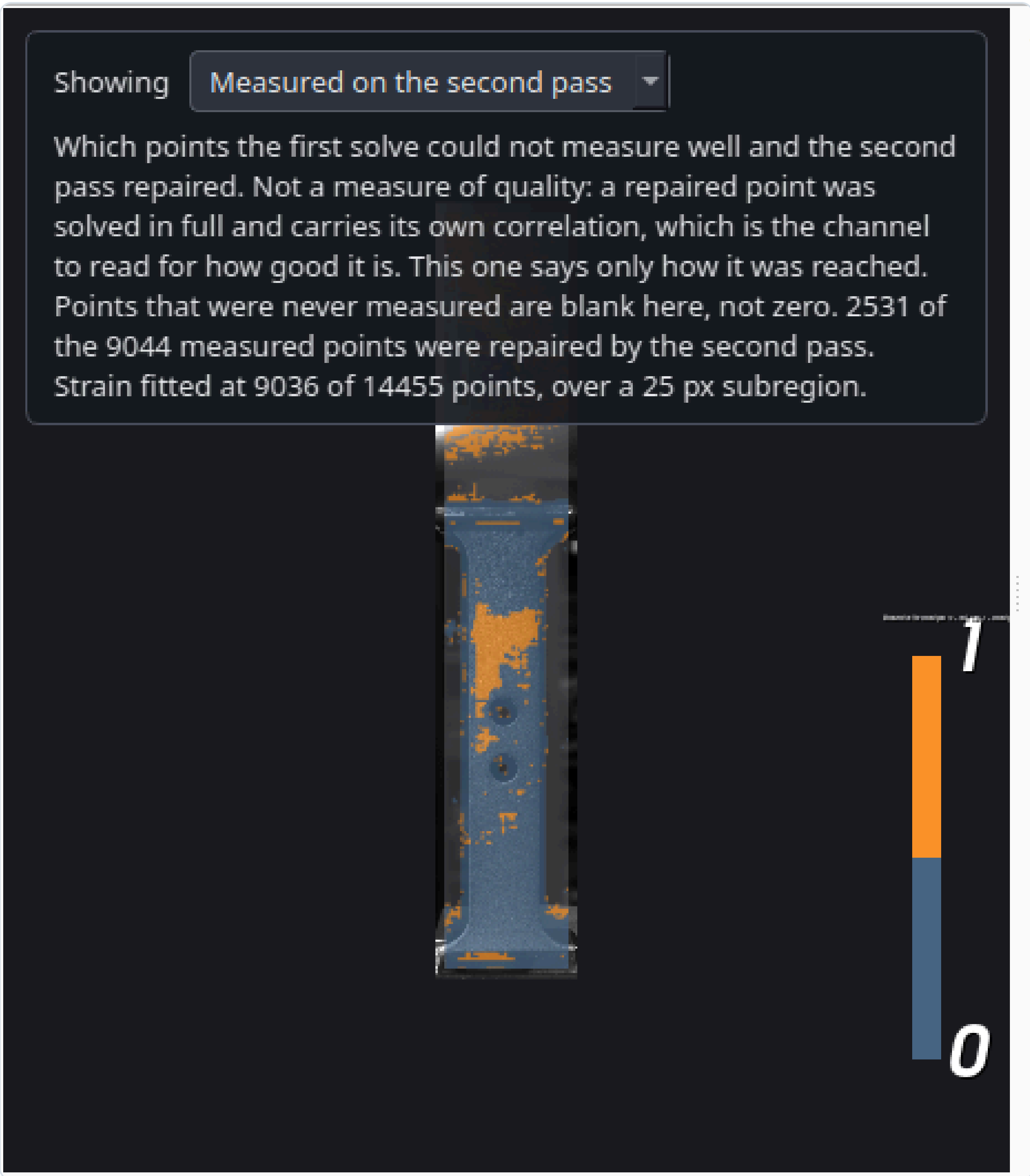
The rule that makes this safe to leave switched on: it may never make anything worse

A recovery pass that only ever touches points that already failed can, at worst, do nothing. That is the entire argument for running it by default: an answer is accepted only if it converges *and* correlates better than whatever was there before - nothing, in the case of a point that had failed outright - so the pass can only add points to a field, never spoil ones that were already there.

Recovery spreads in rounds: points recovered in one round become trustworthy neighbours for the next, so repair can work its way into the middle of a large failed patch that had no good neighbour at all on the very first attempt.

A recovered point is marked, everywhere it is reported

A recovered point is a genuine measurement - it converged, and it carries its own correlation score - and needs no apology on that account. But it reached its answer by a different route than the rest of the field, and that is exactly the kind of thing this manual has already argued should never be hidden: the mark travels with the point everywhere it is reported, so a reader can always ask where the repairs are concentrated. That question has a real answer worth asking - repairs clustered along one edge of a specimen usually mean something about the specimen; repairs scattered evenly usually mean something about the correlation settings.



Where the repairs fell. 2531 of 9044 measured points on this real specimen were recovered by the second pass. The map answers what the count cannot: they cluster around the holes and along the edges, which is the specimen talking, rather than scattering evenly, which would point at the correlation settings instead. Two flat colours and no gradient between them, because the channel has exactly two states and a ramp would imply readings that cannot occur.

What it actually buys, measured

On real specimens the effect is large. A hard frame of a real tension test with two holes through the gauge section went from **5401 points measured to 8455** once the pass was switched on. On a run that already measured most of its field, the same pass costs a modest amount of extra time and changes almost nothing - the effort scales to how much of the field was actually broken.

pair	before	after
synthetic large strain, first to last frame	167	8099
synthetic rotation, first to a later frame	2574	10170
real tension coupon without holes	5401	8455
real tension coupon with holes	6513	9044

9. Two questions, never one score

"How reliable is this measurement" sounds like one question. It is genuinely two, they have different answers, and collapsing them into a single quality score throws away exactly the information that would let you tell which kind of problem you are looking at.

Question one: how well could this ever have gone?

A speckle pattern has more or less texture, more or less contrast, in any given small patch. Some patches are naturally easier to lock onto than others, independent of anything that happens to them afterward - a patch with strong, varied local contrast can be located far more precisely than a smoother, weaker one, purely as a matter of how much information is in the pattern itself.

A **noise floor** answers exactly this: given the pattern's own local contrast and the camera sensor's own noise level, what is the finest displacement this particular subset could ever resolve? It is computed from the reference image alone, comparing the subset's gradient strength against an estimate of the sensor's noise.

> **WHAT IT CANNOT SEE**
>
> Because it never looks at the target image at all, a noise floor cannot see anything that goes wrong *during* the measurement: decorrelation, out-of-plane motion, focus drift, or a shape function too poor to represent the real deformation. It is a lower bound on how good a result could be, never a total error bar on how good the result actually was.

In practice it behaves like a map of speckle quality, expressed directly in the units that matter: pixels of displacement. On real measured data it typically sits in the range of a few thousandths of a pixel - small enough that reading the bare number in isolation is nearly meaningless. Chapter 10 covers how to read it usefully.

Question two: how confident is this particular answer?

Match conditioning asks a different question: around the specific solution this search actually converged to, how sharply does the matching cost rise as the candidate position moves away from it? A sharp, well-defined minimum means the search landed somewhere unambiguous. A shallow, flat-bottomed minimum means many nearby positions matched almost as well, so the reported position, while a genuine best answer, carries more uncertainty about exactly where the true match sits.

Unlike the noise floor, this probes the actual match that was found, so it does respond to problems during the measurement. It is dimensionless and has no absolute scale of its own: it is only meaningful *within* one run, comparing one point against another in the same field, never across two different runs or two different specimens.

Both read the opposite way from everything else

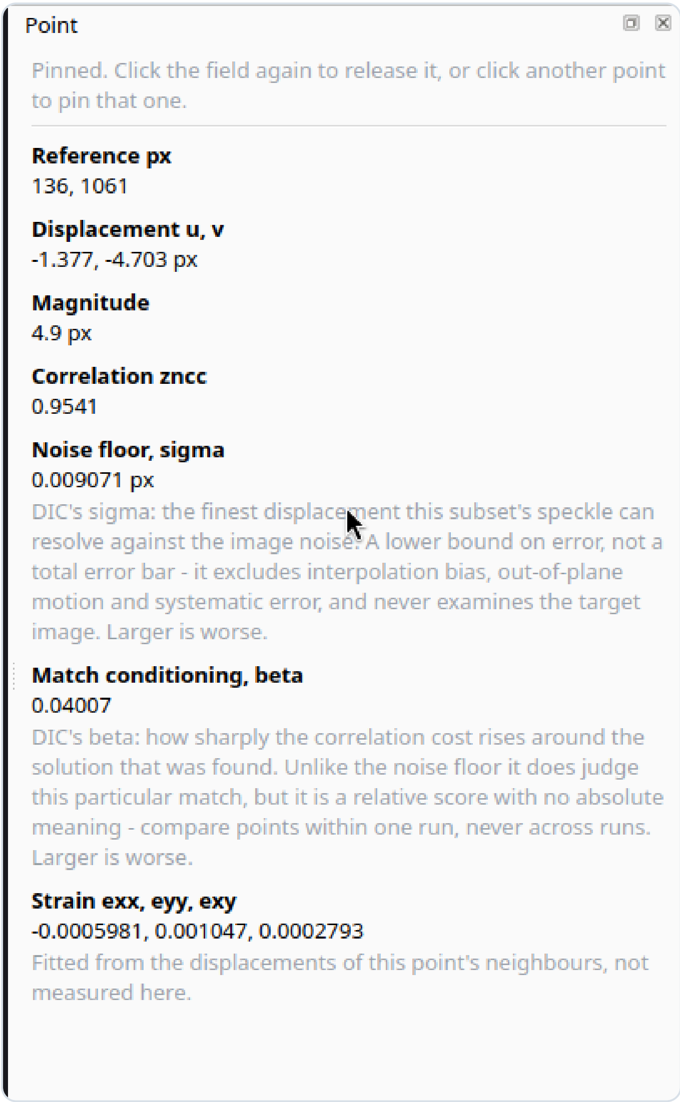
Every other quantity in a DIC result - displacement, strain - is read the ordinary way: a value simply is what it is, neither good nor bad on its own. Both reliability figures are different: for both of them, **a larger number is worse**, the way a margin of error is worse the larger it gets. Reading them with the wrong habit - treating a large value as an interesting hotspot rather than a warning - is an easy mistake to carry over from every other channel, and it needs to be stated plainly wherever these numbers appear, not left to be inferred.

Zero is the most dangerous value either can show

Neither figure can honestly reach zero. A zero noise floor would claim a perfectly resolvable measurement; a zero match conditioning would claim a perfectly sharp, unambiguous cost minimum. Neither is physically reachable. So a value that is not strictly positive was never actually established - it is the same "absence, not zero" principle from Chapter 3, and it applies with particular force here, because zero looks like the *best* possible reading a reader could hope for, rather than an obviously suspicious one.

A bare number needs something to be measured against

A noise floor of a few thousandths of a pixel means nothing on its own - whether that is excellent or barely adequate depends entirely on how large the displacements being measured actually are. Putting it against the largest displacement the run actually measured turns it into a sentence that means something on its own: something like "at worst, one part in several hundred of the largest movement measured." That comparison uses only numbers the run already produced; it invents no external idea of what counts as a "good" noise floor, which would only be a threshold pulled from nowhere in particular.



One point, and what each of its numbers is not. Every channel carries a sentence stating its own limits, next to the number rather than in documentation nobody has open. The noise floor says it is a lower bound and never examines the target image; the conditioning says it is comparable within this run and meaningless across runs.

10. Reading a field critically

A colour map is persuasive by nature - smooth gradients look authoritative whether or not they deserve to. This chapter is a checklist for looking at any DIC field, from any tool, without being talked into more confidence than the data actually earns.

Read the scale before reading the colours

What is the range, and what are the units? A field that runs from 3.000000 to 3.000002 pixels is, for all practical purposes, uniform - and drawn with the same rainbow as a field that genuinely spans a wide range, it can look just as dramatic. The colours alone cannot distinguish real variation from noise sitting at the sixth decimal place; only the number on the scale can.

Is zero centred, or is it not?

A strain scale should be centred on zero, because zero strain is a real physical state and the sign either side of it means something (tension against compression). A displacement scale should not be, because a specimen that merely translated across the frame has no reason for its displacement to straddle zero at all. If a scale's centring does not match which kind of quantity is on screen, treat every colour on it with suspicion until that is understood.

Which direction is "worse"?

For an ordinary measurement, a large value is just a large value. For a reliability channel (Chapter 9), a large value is a warning. The same warm-to-cool colour ramp can be used for both, and nothing about the colours themselves tells you which reading applies - only the label does. Check what is actually being shown before deciding whether the red patch is the interesting result or the part to distrust.

Are the holes really holes?

Ask what an unmeasured point would look like on this particular map. If the honest answer is "identical to a genuine zero," as Chapter 3 warns, then a smooth-looking field may be smooth because the tool filled in the gaps with the most reassuring possible value rather than because the specimen actually behaved smoothly. A field that is willing to show visible holes is, perhaps counterintuitively, more trustworthy in the parts that remain than one that never does.

Does the field extend further than the measurement that feeds it?

A strain map should never reach further than the displacement map it was fitted from (Chapter 2), and a repaired point (Chapter 8) should be visibly marked as one if that information is available at all. A field that looks suspiciously complete, with no trace of where the underlying measurement actually struggled, is a field worth asking harder questions about.

> **THE FIVE-QUESTION PASS**
>
> Scale and units. Centred on zero, or not. Which direction is worse. Are the gaps real. Does this field outrun what it was built from. Running through these five before reading any conclusion off a DIC field takes seconds, and it is where most confident misreadings actually get caught.

11. Reading a sequence as a curve

One field is a snapshot. What a loading test is actually for is a curve - strain against load step, elongation against time - and getting from a stack of fields to a trustworthy curve has its own ways of going wrong.

The virtual extensometer

A physical extensometer is a clip gauge: two arms attached to a specimen a fixed distance apart, reporting how that distance changes as the specimen deforms. A **virtual extensometer** is the same idea done in software: place two points anywhere on the measured field, and read the change in the distance between them across every frame of the sequence.

It measures the gap between the anchors, never the movement of either one

This is the rule that matters most, and it is easy to get backwards without noticing, because a common way to test the idea does not expose the mistake. A gauge has to compute the change in **distance between** its two anchor points, not the displacement of either anchor on its own. A specimen carried bodily across the frame - a pure rigid translation, no stretching at all - has genuinely strained by nothing whatsoever. An implementation that (by mistake) read one anchor's own displacement instead of the gap between the two would turn that rigid motion into a large, smooth-looking, entirely fictitious strain reading.

The trap in verifying this: testing only with a pure stretch centred on the origin will not catch the bug, because under that specific kind of deformation the two calculations happen to agree by coincidence. It takes a genuinely rigid translation, tested on its own, to tell the two apart.

An anchor needs real measurement on all sides to mean anything

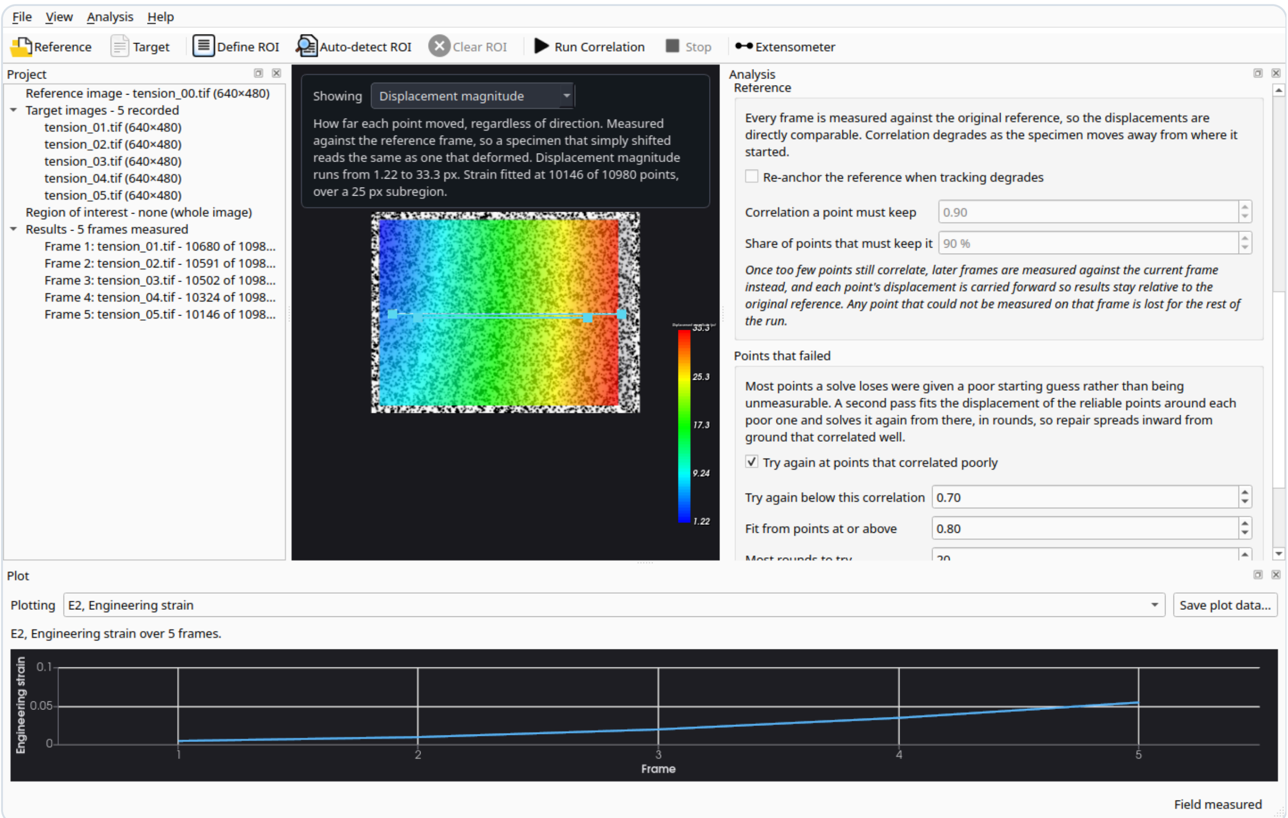
An anchor placed between grid points does not sit exactly on a measured point; its reading is interpolated from the four points around it. If any one of those four is a gap (Chapter 3), every obvious way to paper over it is dishonest in its own way: using the three that remain quietly changes what is being averaged; reaching further out for a substitute borrows a reading from somewhere the anchor is not; filling the gap with zero drags the whole reading toward "did not move," which is the most plausible-looking wrong answer available. The honest response is to refuse the reading outright and let the anchor be moved somewhere the field can actually support it.

A gap in the curve is drawn as a gap

Some frames will not be readable - an anchor sat over a hole (Chapter 5), or that frame's field simply did not measure enough of the specimen. Dropping that frame from the curve and drawing a straight line from the frame before it to the frame after would fill territory nobody actually measured with a segment that is, on the screen, indistinguishable from real data. The honest curve breaks visibly where a frame could not be read, and says so in words, rather than smoothing over the hole.

Verified, not merely argued

A synthetic tension sequence ships with an exactly known answer for every frame, and a virtual extensometer placed on it reads 0.005, 0.010, 0.020 and 0.036 against stated strains of 0.005, 0.010, 0.020 and 0.035 across four frames - close enough, over displacements reaching tens of pixels, to trust the method rather than merely the argument for it.



A loading curve from two clicks. The extensometer is the pale line drawn across the specimen; the curve beneath reads the change in the distance between its two ends across every frame. The caption above the chart states how many frames were readable, so a break in the line can never be mistaken for smoothing.

12. The field leaving the application

A result rarely stays where it was produced. It gets opened in another tool entirely - a general-purpose scientific viewer, a spreadsheet, a finite-element package - and every principle this manual has argued for has to survive that trip, or it was never really upheld at all.

Geometry says where the instrument looked; the data says what it found

An exported field's underlying mesh should cover every place a measurement was *attempted*, not merely every place it *succeeded*. If the geometry itself shrank to fit only the successful points, a difficult specimen would quietly export as a smaller, easier-looking one than it really is - the very difficulty that makes it worth studying would disappear from the file before anyone else ever saw it. The data arrays sitting on that geometry are where "attempted but not measured" actually gets stated, following the same absence-not-zero rule as everything else.

The same "not zero" rule, one file format further out

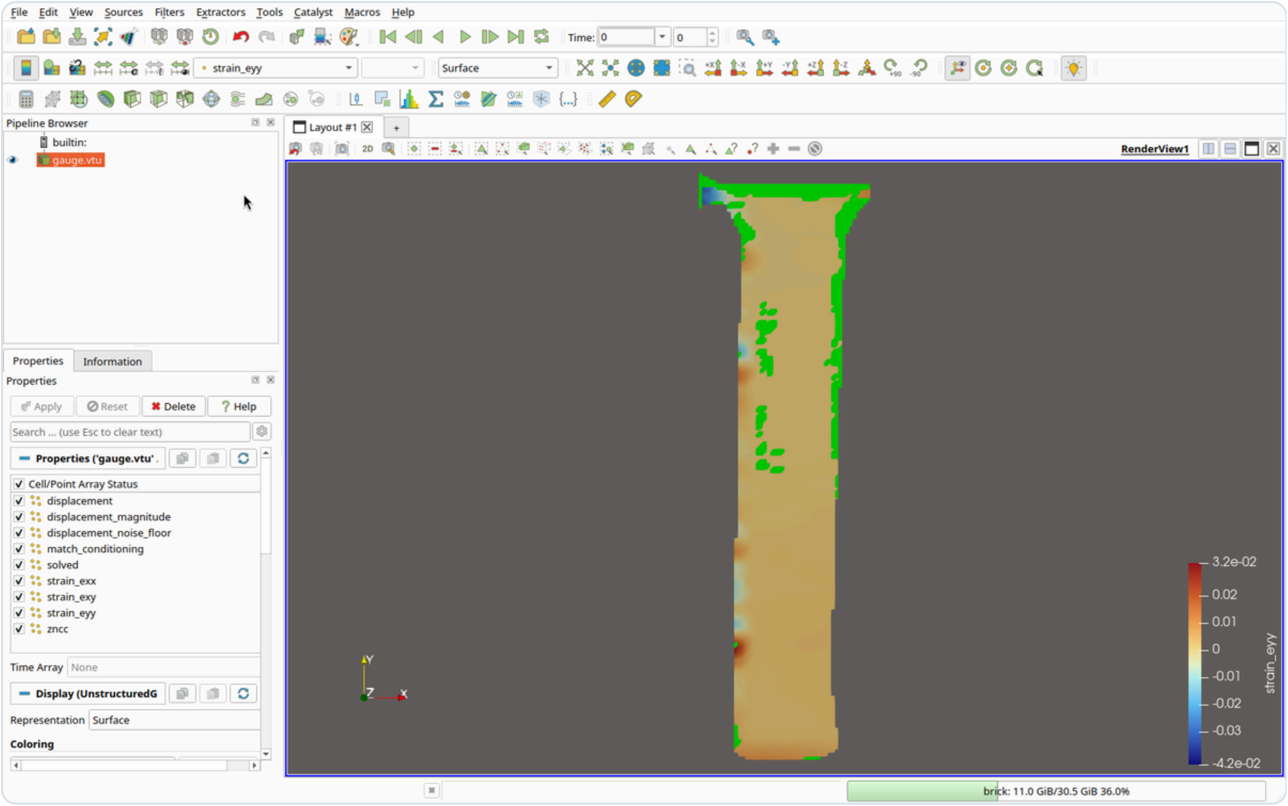
In almost any viewer, a field of zeros renders as a perfectly reasonable, uniform blue region - entirely indistinguishable from a genuine measurement of no movement. An export has to carry unmeasured values as not-a-number, exactly as the application's own screen does, and state outright, in an array of its own, which points were actually measured - rather than leaving a reader to infer it from wherever the holes happen to be. A derived quantity that was never asked for at all (strain, on a run where it was switched off) should be *absent* from the file entirely, not present and filled with not-a-number everywhere: an array that is entirely not-a-number reads as "we tried and failed everywhere," which is a much more alarming and much less accurate claim than "this was never attempted."

A file has to state its own coordinate frame

Chapter 4 explained why one consistent coordinate frame matters inside the application. The moment a field leaves it, that guarantee is gone: whatever opens the file next has its own default assumption about which way is up, and there is no reason to expect it matches. The file has to say, explicitly, which convention its coordinates use, so a reader - human or software - does not have to guess and does not draw the field mirrored without any way to notice.

Provenance travels with the file, captured at the moment that matters

A trustworthy result carries an account of how it was made: both source images identified precisely enough that a substitution would be caught, the settings the correlation actually ran with, the region that was used, and the exact version of the software and its engine that produced it. That account has to be captured **when the run starts**, not read back from an on-screen panel afterward - a panel that keeps accepting input after a run finishes will, if read late, describe a configuration that never actually produced anything, attached with complete confidence to a result that was really made under different settings entirely.



The same field, opened somewhere else. No conversion step: the format is VTK's own, so ParaView and FreeCAD's FEM workbench read it directly. Every channel the run measured is in the array list, and the gaps are still gaps - unmeasured points arrive as not-a-number rather than as a flat blue region no reader could tell from a real measurement of no movement.

A TRAP WORTH KNOWING ABOUT, GENERALLY

A file-writing call that reports success is not automatically proof a file was written. It is worth checking, once, that the specific writer a pipeline depends on actually fails loudly - some report success even when the destination directory does not exist and nothing was written at all. A confirmed write, not merely an unrejected one, is the only thing that should tell a user their result is saved.

13. Speckling a specimen

Chapter 1 explained why a speckle pattern is necessary at all: a featureless surface has no unique texture to lock onto. This chapter is about what makes one pattern actually work well and another one fail quietly.

What a good pattern needs

- **Random, not periodic.** A repeating pattern - an evenly spaced grid of dots, a woven fabric texture - gives the correlation search more than one place that looks like an equally good match. The search can lock onto the wrong repeat of the pattern with high apparent confidence, which is a far worse failure than an honest low correlation score: it reports a plausible, wrong answer instead of admitting doubt.
- **High local contrast.** The correlation is only as good as the intensity gradient inside each subset. A washed-out, low-contrast pattern gives the search little to grip, which is exactly what a noise floor (Chapter 9) measures directly: weak local contrast against a given camera's own noise level produces a correspondingly coarse noise floor, however good the pattern looks to the eye.
- **Feature size matched to the subset, not to the whole image.** What matters is not how the pattern looks across the whole specimen, but how much distinct texture falls inside one subset. A pattern with speckles far larger than the subset gives each subset too little variation to be unique; a pattern with speckles far smaller can blur together at the imaging resolution actually being used. Chapter 14 covers choosing the subset itself; the two decisions have to be made together, not in either order alone.
- **Genuine full-range contrast, without saturating.** A pattern that is mostly pure black and pure white loses gradient information exactly where the camera clips - a saturated pixel carries no information about how much brighter or darker the true value was, so a pattern that clips throws away contrast it could otherwise have used.

How a pattern gets onto a specimen

Some specimens already carry enough natural texture to correlate well and need nothing applied at all - worth checking, with a real photograph and a real quality estimate, before assuming a pattern is necessary. Where one is needed, common methods include spray or airbrush application of a fine random speckle, and printing a computer-generated, non-periodic pattern onto a decal or stencil and transferring or applying it to the specimen.

Checking a pattern before committing to a whole test

The cheapest check is the one done before any load is applied at all: photograph the specimen as speckled, and look at the reliability figures a correlation of that reference image against itself, or against a first small step, actually produces. A live estimate of what a chosen subset radius can resolve against the pattern actually present is worth having on screen while those settings are still being chosen, precisely because it is far cheaper to re-speckle a specimen before a test than to discover a pattern was too weak only after the test is finished and cannot be repeated.

14. Choosing subset radius and grid step

Two numbers control almost every correlation: how large a patch of pixels each measurement point looks at, and how far apart the measurement points are spaced. Neither has a single correct value; both are trade-offs that depend on the pattern and on what is being measured.

Subset radius: robustness against locality

A larger subset contains more texture, which generally makes its correlation more robust and its noise floor finer (Chapter 9) - more pixels means more gradient energy to work with, and a steadier average against sensor noise. The cost is spatial: a large subset blurs the field, because a single reported point is really an average over a fairly wide patch of the specimen. Where the deformation changes sharply over a short distance - near a hole, at a crack tip, across a narrow gauge section - too large a subset averages the interesting detail away before it is ever reported.

A smaller subset can follow sharper local detail, but has less texture to work with, is more sensitive to noise, and needs a correspondingly finer speckle pattern to still contain enough unique contrast to correlate reliably at all.

Grid step: sampling density against cost

The grid step is simply how far apart the measurement points are placed. A finer step gives a denser field and a smoother-looking result, at a roughly proportional cost in computation. It does not, on its own, improve the quality of any individual point's measurement - that is entirely the subset's job - so a very fine grid step over an unreliable subset produces a dense field of unreliable answers, not a genuinely more precise one.

Working from the reliability figures, rather than guessing

The two settings are not really independent, and neither is right or wrong in isolation: the sensible way to choose them is to look at what the actual pattern, at the actual imaging resolution being used, can resolve at a candidate subset radius - the noise floor from Chapter 9, computed live against the settings under consideration - rather than picking a subset radius by habit or by copying a value from an unrelated test.

15. Lighting and imaging

Everything in Parts I through IV assumes the two photographs differ only by the deformation being studied. Lighting, focus, and camera settings that drift between reference and target introduce differences that have nothing to do with the specimen, and DIC cannot tell those apart from real movement.

Even, diffuse, and stable

Direct, hard lighting produces specular highlights - small, very bright patches that shift position as the specimen itself moves, even though the surface underneath is speckled normally. A highlight sliding across a subset between reference and target looks, to a correlation search, like a change in the pattern itself, and can corrupt an otherwise perfectly good measurement. Even, diffuse lighting, without a strong single source, avoids this.

Lighting also has to stay **stable** for the whole duration of a test, not merely at the moment each photograph is taken. A light that dims, flickers, or casts a moving shadow across the specimen over the course of a long test introduces intensity changes a correlation search has no way to separate from real deformation.

Fixed focus, fixed aperture, fixed everything that can auto-adjust

Any camera setting that can change automatically between shots - autofocus, auto-exposure, auto white balance - is a setting that can quietly change the image in a way indistinguishable, to the correlation, from real deformation. All of them should be fixed manually before a test begins and left untouched for its entire duration, including through however many load steps and however much time the test takes.

The camera itself has to hold still

Everything this manual has said about a stable coordinate frame (Chapter 4) assumes the camera did not move between the reference photograph and any target. A tripod or rigid mount that cannot shift, even slightly, over the course of a test is not a minor convenience - a camera that moves introduces a rigid displacement across the entire field that is real, in the sense that it genuinely happened, and completely uninteresting, in the sense that it has nothing to do with the specimen being studied, and no way to tell the two apart after the fact.

> This chapter covers the fundamentals that apply regardless of equipment. It does not yet cover camera and lens selection, or the calibration procedure a stereo rig needs - that workflow is on SurView's own roadmap and will get a chapter here once it exists in the application. See the Appendix for what else is still to come.

APPENDIX

Glossary

Terms as this manual uses them. Where SurView's own screens use a different word for the same idea, that word is given alongside it.

Reference image

The photograph taken before deformation. Every displacement in a run is measured relative to it.

Target image

A photograph taken after some deformation has occurred, compared against the reference (or, after reference updating, against a previous target) to measure movement.

Subset

The small patch of pixels around one measurement point that is matched between the reference and the target. Its size is the subset radius.

Grid step

The spacing, in pixels, between neighbouring measurement points.

Correlation coefficient (ZNCC)

A number stating how closely a candidate match in the target resembles the original subset in the reference. A measure of matching confidence, not of physical correctness.

Displacement field

The measured movement, in pixels, at every point of the grid.

Strain

Local stretching or shearing, fitted from the displacements of neighbouring points rather than measured directly at any one point. Reported in three common forms: Cauchy (small-strain), Green-Lagrange, and Euler-Almansi, which agree closely at small deformations and diverge at large ones.

Region of interest (ROI)

The boundary restricting where measurement points are placed, drawn by hand or proposed by automatic segmentation of the speckle quality. May exclude one or more holes: places inside the outer boundary that are not to be measured.

Sequence

A reference image plus an ordered series of target images, making up one loading test.

Reference updating (re-anchoring)

Switching a sequence's comparison from the original reference to a more recent frame, once too little of the field still correlates against the original. Off by default.

Recovery pass ("Points that failed" in the Analysis panel)

A second attempt at points the first correlation could not measure well, seeded with a displacement fitted from nearby reliable points and then fully re-solved. On by default; never accepts an answer worse than the one it would replace.

Noise floor (sigma)

The finest displacement a given subset's own speckle quality could resolve against the camera's sensor noise. A lower bound, computed from the reference image alone.

Match conditioning (beta)

How sharply the matching cost rises around the solution actually found for one point. Meaningful only within a single run.

Virtual extensometer

Two points placed on the measured field whose changing distance apart is plotted against frame, standing in for a physical clip gauge.

Where to find it on screen

To do this	Look here
Import the reference or target images	Toolbar: **Reference**, **Target**
Draw or auto-detect a region	Toolbar: **Define ROI**, **Auto-detect ROI**
Exclude a hole from a region	Toolbar: **Add Hole** (needs a region first)
Choose the solver, subset, grid step, strain settings	**Analysis** panel
Turn reference updating on and see its cost	**Analysis** panel, "Reference" group
Turn the recovery pass on and adjust it	**Analysis** panel, "Points that failed" group
Start or stop measuring	Toolbar: **Run Correlation**, **Stop**
Read a value at one point of the field	**Point** panel; hover the field, click to pin
Switch between displacement, strain, and reliability channels	The "Showing" selector above the field
Place a virtual extensometer	Toolbar: **Extensometer**, then two clicks on the specimen
Plot a quantity against frame	**Plot** panel
See provenance for an imported image	**Record** panel
Export a field	File menu: `.vtu` for ParaView and FreeCAD, or `.csv`

What is not in this manual yet

Written up only where the screen it describes actually exists, this manual is necessarily behind SurView's own roadmap. As of this writing, the following are real, tracked work that simply has not landed:

- Stereo and out-of-plane (2.5D) measurement, and the camera calibration workflow it depends on.
- Reading a quantity along a line, rather than only at one point or between two extensometer anchors.
- Multi-view measurement from a single moved camera, for slow or static tests such as creep.
- Sparse point-marker tracking, alongside the dense speckle field.

See ROADMAP.md for the current state of each. This manual will gain a chapter for each one as it ships.