3D documentation certification methodology

aPrehliadky · methodology version 2 (September 2026, aPrehliadky 3.0) · forensika.eu

The 3D documentation certificate records how the scan was made (capture trace), what substantiates its scale (verifications) and exactly what it certifies (file hashes). The app issues it for a finished model; it exists as certificate.json in the .f3d package and as a PDF for the case file. The certificate computes nothing — every value in it comes from the session manifest or from a file hash.

The certificate is not a metrological calibration of the device and does not replace expert judgement. It states by how many independent routes the model scale is substantiated, and with what result.

1. Sources of scale and their independence

Source	What it measures	Independent of
LiDAR depth (time-of-flight)	distance of every image point from the camera; photogrammetry receives a depth prior — it does not guarantee the model scale (audit 3.0: the same scan came out 6 % and 14 % smaller)	photogrammetry, the operator
Camera trajectory fit (ARKit → model)	similarity transform between camera positions from ARKit (IMU + visual odometry + LiDAR, metric) and poses from photogrammetry; from 3.0 the primary source of the measurement scale (coefficient 1/k, uncertainty $u(k) = \sqrt{(\sigma/(r\cdot\sqrt{n}))^2 + 0.5\,\%^2}$)	LiDAR depth in photogrammetry, the target
T3 target in the model	the centres of circular fiducials (Ø 24 mm, base 247 × 160 mm) are found in every frame and triangulated from the photogrammetry poses directly in the model frame; 10 distances (33–294 mm) vs. the reference × print scale → k with its uncertainty (~0.1–0.3 %)	LiDAR and the ARKit trajectory; verifies photogrammetry
Reference	the operator measures a segment of known length in the model (100 mm target scale, ruler) and enters the true value	LiDAR, ARKit; depends on the operator
Target (LiDAR) — T2 and T3	the fiducial centres are computed in 3D from LiDAR depth and the camera pose; the distances are compared with the printed reference	photogrammetry, the operator; verifies the LiDAR + ARKit chain

The measurement scale is the fusion of the trajectory and the target in the model (inverse-variance weighting); their difference is the independent check. Every measurement carries an expanded uncertainty U (k = 2) from the scale uncertainty and the point placement (finger 1.7 mm, Apple Pencil 0.5 mm, mouse 0.2 mm, + frame resolution). The T3 target also serves validation of the app and the device: same target, same procedure, many scans and devices → the error distribution is material for the validation study (docs/VALIDACIA_protokol.md). The print is verified with a caliper (centres A–B = 247.00 mm) and the value is entered in the app.

2. Documentation class

Class	Conditions	Meaning
A — metrologically verified	LiDAR depth for ≥ 90 % of frames and scale corrected by a metrology-grade trajectory fit (residual < 25 mm, ≥ 50 cameras) and independent verification (reference, LiDAR target or target in the model) with error ≤ 1 % (or two independent verifications)	measurements substantiated by two independent routes; values binding within the uncertainty U

Class	Conditions	Meaning
B — metric	scale corrected by a metrology-grade trajectory fit or LiDAR scale confirmed by a reference/target; independent verification missing or ≤ 2 %	the model is metric, a second proof is missing — state “no reference” in the report
C — indicative scale	scale known only approximately: LiDAR without a fit correction (really a 6–14 % error) or a coefficient from the trajectory without LiDAR; verification ≤ 5 %	values indicative; add a reference before citing in an expert opinion
D — unverified scale	scale not substantiated (no LiDAR and no trustworthy fit) or verification > 5 %	illustrative documentation only

“Trustworthy fit” = residual / scale < 150 mm, ≥ 8 cameras (enough for cropping). “Metrology-grade fit” = residual / scale < 25 mm, ≥ 50 cameras (enough to correct the scale; real scans: 7.6–8.5 mm with 158–198 cameras). “Independent verification” = the smallest of the reference error, the LiDAR target error and the difference between the target in the model and the trajectory; the LiDAR target counts with ≥ 3 distances and ≥ 5 observations per fiducial. The class is computed by the same function in the report and in the certificate, so the two documents are always consistent.

3. Capture trace

From the frame metadata (camera pose, time, depth) the certificate states: start and end, duration, number of frames and frames with depth, resolution, coverage (azimuths of 36, elevation bands of 3), camera path length, range and mean of the camera distance from the scan centre, number of frames with a validly found target. The PDF shows a top view: camera path, frames, scan centre and the 3D fiducial positions. The trace shows that the scan was captured according to the procedure (three heights, full circle, appropriate distance) — and exposes one that was not.

4. Certified content and integrity

- SHA-256 of the model (model/model.usdz), the room mesh (model/room.usda) and the findings (annotations.json) at the time of issue; number of findings and measurements.
- SHA-256 of the archive package, if the scan was archived.
- payloadSHA256 = SHA-256 of the canonical certificate bytes (sorted keys, ISO 8601 dates, no whitespace). Any change to the certificate invalidates the hash.
- Signature (Pro): ECDSA P-256 / SHA-256 with the device key (Secure Enclave where available) over the canonical bytes; the certificate carries the public key (X9.63), the signature (DER), time, signer, institution and the key fingerprint (first 8 bytes of the SHA-256 of the public key).
- Identifier F3D-XXXXX-XXXXX = SHA-256(session ID + model hash + issue time). QR code in the PDF: F3D:<ID>:<class>:<model hash prefix>:<session ID>.

5. Verifying a certificate

1. In the app: import the .f3d package. The import verifies the hashes of all files, the package signature (if any) and the certificate: content integrity, the match of the model hash with the model in the package and the certificate signature. The result is in the session overview (“Certificate in package — valid”) and in the report.
2. Without the app: unzip -p scan.f3d model/model.usdz | shasum -a 256 must give the hash from the certificate; unzip -p scan.f3d certificate.json contains the certificate, its payloadSHA256 and the signature; any ECDSA P-256 library verifies the public key and signature over the canonical bytes.
3. Compare the QR code in the PDF with the ID and hash in the package.

6. Device validation — recommended procedure for a study

1. Print the T3 target at 100 % (A4 landscape), mount it on a rigid board, check the circle centres A–B (247.00 mm) with a caliper and enter the value in the app; a print deviation > 0.5 mm → new print.
2. Target flat, matte background, diffuse light. Capture ≥ 10 scans with the Object profile, extent Detail, distance 0.3–0.6 m, three heights, on each device.
3. From each scan export the Validation CSV (Export → Validation CSV): scale and uncertainty from the trajectory, from the target in the model and their difference, the LiDAR target error, sharpness, class and all measurements with U.
4. Evaluate mean, SD and the 95th percentile of errors by device and distance; the class A criterion (≤ 1 %) should be met in ≥ 95 % of scans.
5. Validate larger dimensions (body length) with a 500–1000 mm reference (measuring tape, two marks) — reference in the report.

7. Limitations

- LiDAR depth has a resolution of 256 × 192 px; the fiducial centre accuracy is ~1–3 mm at 0.5 m and decreases with distance and oblique views. Hence the median over many frames and outlier removal.
- The LiDAR target verifies the LiDAR + ARKit chain; the scale of the photogrammetric model comes from the trajectory fit and the target triangulated in the model (fusion). The certificate reports all routes separately.
- The colour patches of the target are indicative; colour-binding documentation requires a commercial ColorChecker.
- The certificate confirms the state at the time of issue; later findings and measurements change the findings hash — issue a new certificate after edits.