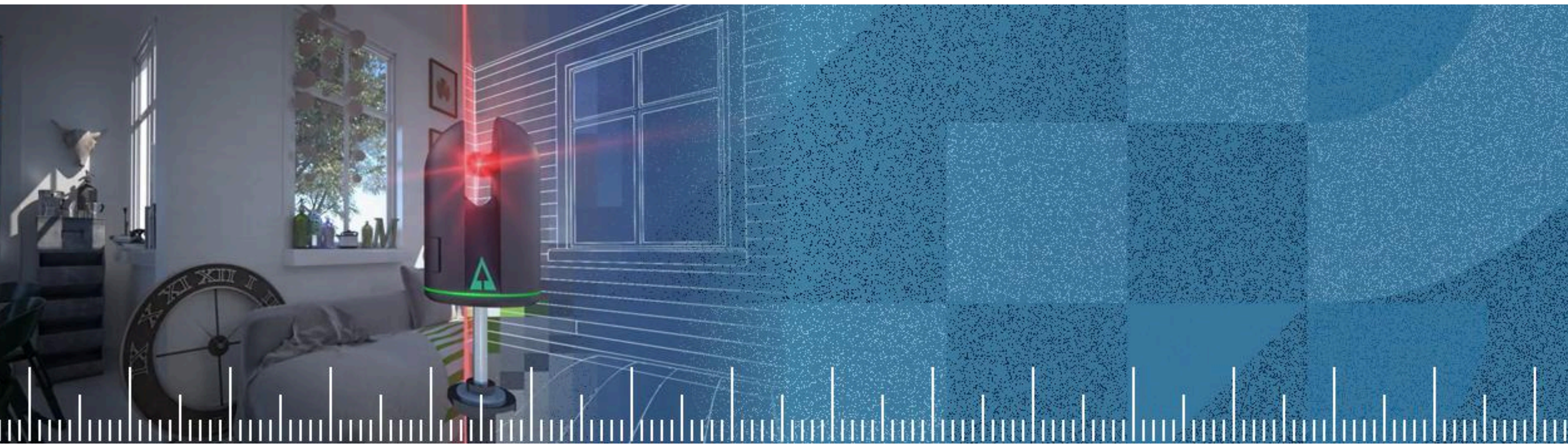
DepthPano

An operational image format delivering web-native residential scans



Metrological Research Foundation

Authors: Nathan Walker, Jim Bosdriesz

June 2026

Contents

Abstract

This paper describes DepthPano, the depth-image format the Metrological Research Foundation uses for indoor terrestrial laser scan data in residential and small-commercial settings. The format encodes radial depth as a 24-bit fixed-point value across the RGB channels of a PNG image at equirectangular projection, with a per-image scaling factor stored in a tEXt chunk, partnered with separately-stored black and white (B&W) reflected-laser-intensity and colour-camera panoramic images at the same projection. Mapping point cloud data onto 2D image grids is an established technique in computer vision. The paper is offered as an operational engineering description of one application of that family, tuned for residential accessibility assessment under Australia's National Disability Insurance Scheme (NDIS) Complex Home Modifications (CHM) pathway and integrated with the Foundation's broader bundle architecture.

End-to-end file-size reductions of approximately twenty-five-fold relative to the equivalent E57 representation are reported. A typical 230 MB E57 produces a complete DepthPano bundle of roughly 8.5 to 9.5 MB. The reduction comes mainly from the structural choices the architecture makes (implicit coordinates, downsampling to the canonical resolution, separation of colour and intensity into paired panoramas) rather than from compression of the depth values themselves; PNG's Deflate contributes a further 38–48% on the bit-packed depth bytes. The format supports direct array-indexed spatial queries on the CPU, avoiding the Spherical-to-Cartesian conversion that point cloud formats require for equivalent operations. Published alternatives achieve better compression (the differential encoding of Kohira and Masuda, 2017) or better-performing projections (Dumić et al., 2022); the Foundation's choice of equirectangular projection is driven by integration requirements with the rest of the bundle architecture, not by independent geometric or compression advantages.

About this paper

This paper describes DepthPano, the depth-image format used in the Metrological Research Foundation's spatial data architecture for indoor terrestrial laser scanning in residential and small-commercial settings. The format is presented as an engineering description: what is implemented, how it works, what trade-offs were accepted, and what file-size and query-performance results are achieved in operational use.

The paper is not a theoretical contribution. Mapping 3D point cloud data onto 2D image grids is a well-established technique in computer vision; the MPEG V-PCC codec, the differential PNG approach of Kohira and Masuda (2017), and the equirectangular comparison work of Dumić et al. (2022) are among the published works that have explored variations of the same architectural idea. The Foundation's DepthPano implementation is one application of this architectural family, with engineering choices made for the specific operational context of NDIS-funded residential accessibility assessment in Australia.

The contribution of this paper is the documented implementation: the encoding details, the integration with the rest of the Foundation's bundle architecture, the operational results, and the direct comparison with the published alternatives. Readers looking for a novel compression algorithm or a new spatial data structure should consult the cited prior work; readers looking for a working format applied at scale in a specific domain may find the present description useful.

The format is published in the public domain under open access terms. The Foundation has no patent claims on the format itself. Adopters are welcome to use, modify, or extend the format without restriction; the paper is offered as documentation rather than as a basis for licensing.

1. Introduction

Terrestrial Laser Scanning (TLS) produces dense, accurate measurements of indoor and outdoor spaces. In the Foundation's operational context, residential accessibility assessment under NDIS, a typical three-bedroom dwelling scan project covers 10 to 15 scan positions per dwelling and produces several gigabytes of raw data per dwelling when stored in standard formats such as LASer file format for point cloud data (LAS; LAZ is its compressed form) or 3D point cloud data exchange format defined by ASTM E2807 (E57).

These data volumes constrain the delivery and review workflows that the Foundation's research program is concerned with. Tong and Bosdriesz (2026) documented the clinical-to-construction handover gap in the CHM pathway and argued that scan-based evidence could reduce inter-rater variability and improve handover fidelity. Realising those benefits requires that the scan data be deliverable to occupational therapists (OTs), project managers, National Disability Insurance Agency (NDIA) reviewers, and tribunal-appointed experts without specialist software and without prohibitive bandwidth or storage costs.

The standard responses to this constraint are point cloud specific compression formats (LAZ, Draco) and proprietary cloud-based viewers. Both have limitations. Compressed point cloud formats reduce file size but retain the requirement for point cloud aware decoder software, which is generally desktop-based and licensed. Proprietary cloud viewers solve the delivery problem but introduce vendor dependence and re-encode the data in ways that are not transparent to the downstream consumer.

DepthPano is the Foundation's response to this constraint for indoor scanning. The format maps the depth information from a fixed-position TLS scan onto a 2D equirectangular image, with depth encoded as a 24-bit fixed-point value distributed across the RGB channels of a PNG file. The separately-stored B&W reflected-laser-intensity and colour-camera panoramic images are held at the same projection. Both files are web-native and can be displayed, processed, or transmitted by any system that handles standard image formats such as JPEG and PNG.

The architectural idea of mapping point cloud data to a 2D image grid is not novel. The MPEG V-PCC codec uses 2D mapping as the basis for point cloud video compression. Kohira and Masuda (2017) demonstrated that PNG-based 2D representation of point cloud differentials achieves competitive compression for mobile mapping data. Dumić et al. (2022) evaluated equirectangular projection as one of several 2D mappings for static point cloud compression. The present work is an application of this established family of techniques, tuned for the operational characteristics of indoor residential scanning and integrated with the Foundation's broader bundle architecture for verifiable spatial evidence.

The paper is structured as follows. Section 2 describes the DepthPano format itself: the projection, the encoding, the metadata, and the wider Measurement stack. Section 3 states the trade-offs and the limitations explicitly. Section 4 describes how spatial queries (distance, angle) are computed from DepthPano data, with the recognition that the operational benefit is the avoidance of Spherical-to-Cartesian conversion rather than algorithmic complexity reduction. Section 5 reports test results comparing DepthPano against E57 on the Foundation's operational corpus. Section 6 compares the implementation against the published prior work. Section 7 concludes. Appendix A documents the image generation and downscaling pipeline; Appendix B specifies the bit-packing encoding in detail; Appendix C describes the E57 scanned point data. Appendix D provides a comparison between image formats used by measurements other than Depth data.

2. The DepthPano format

2.1 Projection

DepthPano uses standard equirectangular projection. A 3D point at radial distance r, azimuth θ, and elevation φ from the scanner origin maps to a 2D pixel at coordinates (u, v) given an origin at the top left corner of the image according to:

$$u = \theta / 2\pi \quad (1)$$
$$v = 0.5 - \varphi / \pi \quad (2)$$

Both u and v are normalised to the range [0, 1] and are scaled to pixel coordinates by multiplication with the image dimensions. The Foundation's standard resolution is 3142 × 1571 pixels; a hi-res variant at 6284 × 3142 is used for selected scans where additional precision is required.

The choice of these specific dimensions is not arbitrary. The horizontal dimension of the standard resolution is 2π×500 rounded up to the nearest integer and the vertical is π×500 rounded up; the hi-res dimensions are defined as exactly double the standard dimensions, which coincides with 2π×1000 and π×1000 rounded up. The resulting angular scale is:

- Standard (3142 × 1571): a nominal 0.002 radians per pixel: two milliradians per pixel.
- Hi-res (6284 × 3142): a nominal 0.001 radians per pixel: one milliradian per pixel.

At a typical indoor scan distance of three metres, one milliradian subtends approximately three millimetres of arc on the scanned surface. A standard-resolution pixel therefore resolves approximately six millimetres at three metres; a hi-res pixel resolves approximately three millimetres at the same distance. The precision is appropriate for the NCC dimensional tolerances that govern routine accessibility assessment. The difference between an 820 mm doorway and an 817 mm doorway is roughly the scale of one to two hi-res pixels at typical scan distance.

The π-based dimensions have three further engineering consequences worth noting. First, the angular scale can be derived from the resolution without consulting external documentation: a reader who notices that 3142 is approximately π×1000 has effectively reverse-engineered the projection convention. For a format intended to outlast the software that produces it, this self-documenting property is useful. A file recovered from an archive in twenty years, with the original specification lost, is still interpretable from the file dimensions alone.

Second, the pixel-to-angle conversion is a multiplication by an exact decimal (0.001 or 0.002 rad/pixel by convention) rather than a division by 2π. The constants are exactly representable in IEEE 754 floating-point arithmetic, so the pixel-coordinate-to-angle mapping is reproducible across implementations. Because the dimensions are rounded up, the per-pixel scale is nominal rather than exact. An image of 3142 columns at exactly 0.002 rad/pixel spans fractionally more than the full circle, with the residual amounting to less than one pixel at the azimuth seam. The convention treats the scale as exact and accepts this sub-pixel overshoot. Downstream computation that involves transcendental functions (sin, cos, sqrt) may diverge slightly between implementations of those library functions; the reproducibility property applies to the projection inversion specifically, not to all subsequent operations.

Third, the relationship between standard and hi-res is an exact 2×2 integer operation in each dimension, with no fractional resampling, no anti-aliasing decisions, and no resampling kernel choice required when converting between the hi-res and the standard format. A standard bundle and a hi-res bundle of the same scan have a deterministic relationship; hi-res scans of specific positions can be overlaid against a standard-resolution dwelling-scale model without any reprojection step, because both use the same calibrated grid. Implementation detail on the downsampling step is in Appendix A.3.

2.2 Depth encoding

The depth value at each pixel is encoded as a 24-bit unsigned fixed-point integer distributed across the Red, Green, and Blue channels of the PNG file. The Alpha channel is not used for depth encoding in the current implementation.

The encoding scheme is:

$$d\_int = R \times 2^{16} + G \times 2^{8} + B \quad (3)$$
$$d = d\_int \times scale \quad (4)$$

where R, G, B are the 8-bit channel values for the pixel (in the range 0 - 255), d_int is the reconstructed 24-bit integer, scale is a per-image scaling factor stored in a PNG tEXt ancillary chunk, and d is the radial distance in metres.

The scaling factor is computed per-image from the maximum observed distance in the scan: once the depth array is populated, the maximum observed distance is divided by the largest value representable in 24 bits (16,777,215), and the result is the metres-per-integer-step scale, so the full integer range spans exactly the observed depths (the procedure is specified in Appendix A.6). For illustration, a scaling factor of 1.0 / 2^{23} metres per integer step gives a depth resolution of approximately 0.12 micrometres at the cost of a maximum encodable distance of approximately 2 metres. For the Foundation's operational scans, where the maximum observed distance typically falls between 5 and 55 metres, the per-image scale yields depth resolution in the low micrometre range. This is far finer than the precision of the BLK360 device (approximately 4 mm at 5 metres), so the encoding is not the limiting factor on accuracy.

The depth precision is therefore not fixed at sub-millimetre by the encoding alone; it depends on the scaling factor computed for the image and on the underlying device precision. The encoding is capable of sub-millimetre precision for short ranges but does not guarantee it. The tEXt chunk records the scaling factor used, so a decoder can reconstruct the depth values exactly.

2.3 Image-level metadata

The PNG file carries the following metadata in tEXt ancillary chunks:

- Scaling factor (metres per integer step)
- Scanner identifier and capture timestamp
- Encoding version (currently 2.0)

In addition, the PNG stores a sentinel RGB value for missing values in the tRNs (transparency) chunk. The sentinel value is operationally important. A laser pulse may fail to return (because it hit a non-reflective surface, exited an open window, or was absorbed) and the corresponding pixel must encode this absence rather than a spurious depth value. The Foundation's current convention is to set such pixels to all-zero (R=G=B=0, d_int = 0), which corresponds to a depth of zero metres: a physically impossible measurement that is unambiguously interpretable as a missing sample. Adopters of the format are advised to handle the sentinel explicitly rather than treating it as a literal zero distance.

2.4 The Measurements stack

The Measurements stack is a set of TLS point measurements stored as a layered stack of panoramic images. The present paper specifies the Depth, Laser, Colour and (optional) Heat encoding only. Additional layers, such as the surface-normal panorama, are not primary measurements, instead they are derived by extrapolating from the Measurement stack to create the Space stack. NB: this will be covered in the SpaceGen paper to follow.

The depth image is the primary layer of a small set of co-registered panoramas the Foundation captures at each scan position, referred to collectively as the Measurements stack. Every layer shares the same equirectangular projection and resolution, so the layers are co-registered pixel-for-pixel: a given pixel addresses the same physical direction in all of them. The depth values are derived from the laser return, so although the laser-intensity and colour panoramas are captured first in the device's acquisition sequence, the stack is presented here depth-first, mirroring the field order of the source E57 point record. This lists position (X, Y, Z) before intensity and colour (R, G, B); see Appendix C. The measured layers, in that order, are:

- **DepthPano**
  Radial depth, encoded as a 24-bit fixed-point value across the RGB channels.
- **LaserPano**
  The reflected laser-return intensity, a black-and-white panorama stored as either an 8-bit or 16-bit greyscale image.
- **ColourPano**
  The visible-light colour panorama, a 24-bit RGB image.
- **HeatPano** (optional)
  A thermal panorama, 14-bit data stored in a 16-bit PNG. It is present only for capture devices with an integrated thermal sensor (the BLK360 G1 includes one; the G2 omits it), so a stack may contain four measured layers or three depending on the device.

The single-channel intensity and thermal readings fit comfortably in 16-bit greyscale, whereas depth requires the full 24 bits for its dynamic range; this is why the stack mixes bit depths across its layers, because the Intensity and Heat layers do not have the same precision as Depth and Colour. Additionally, the standard Colour, Intensity and Heat images also can be compressed more aggressively: lossy compression of those layers can reduce their size substantially, while the depth layer must maintain its lossless format to preserve its precision.

Whilst the DepthPano is always stored as a PNG file for lossless compression, and hence accuracy for distance calculations, the other panoramas are data agnostic for format. In general the other images should be stored JPEG files to reduce file size with little impact on quality. However, if the LaserPano or HeatPano images are used for calculations they should be stored as PNG files to preserve their accuracy. For a comparison of file sizes for the alternative file formats consult [Appendix D].

2.5 LaserPano: the reflected-intensity panorama

Depth and intensity both derive from the same laser return, that is, the range to the surface and the strength of its reflection respectively, and the format keeps them as separate co-registered layers rather than packing them together. This mirrors the source E57 point record, where range and intensity are distinct per-point fields (see Appendix C), and it lets each layer retain its own native bit depth.

The laser-return intensity is stored as a separate panoramic image; the LaserPano: a single-channel (black-and-white) image recording the strength of the reflected laser pulse at each point, at the same equirectangular projection and resolution as the depth image and stored as a 16-bit greyscale. Because it records reflectance independently of visible-light appearance, the LaserPano is capable of supporting material discrimination; distinguishing paint, glass, metal and fabric; which could prove useful for identifying features beyond the visible spectrum, e.g. shiny vs matt surfaces.

As the IntensityPano follows a fairly predictable shading distribution, i.e. dark towards the corners and bright square on to the TLS; the JPEG file format is recommended as the best trade-off between file size and quality.

2.6 ColourPano: the paired colour panorama

The depth image carries no colour information. The corresponding colour data is stored in a separate file at the same equirectangular projection and resolution. The two files are co-registered: pixel (u, v) in the depth image corresponds to the same physical direction as pixel (u, v) in the colour image.

This separation is deliberate. Combining colour and depth in shared 8-bit channels would force either colour quantisation (limiting visual quality) or depth quantisation (limiting precision); storing them as separate, co-registered layers removes this coupling, so each can be encoded on its own terms (§2.4). The cost is an additional file in the stack and the manifest entry that goes with it.

The stack is data agnostic on the format of the file used for the ColourPano. JPEG is recommended as the best trade-off between file size and quality.

3. Trade-offs and limitations

The format's limitations are stated explicitly. None of the following are deficiencies; they are properties of the chosen architecture that adopters need to understand.

3.1 Single-return only

A laser pulse may return from multiple surfaces along its path: a window followed by the wall behind it, foliage followed by the ground beneath it, glass surfaces with reflections off both the front and the back. DepthPano's 2D structure can encode only one depth value per pixel direction and therefore retains only one return.

This limitation applies to DepthPano as it would apply to any single image 2D representation. It is sometimes presented as a unique cost of the format, but it is shared with the per-setup view of an E57 file: an E57 representing a single fixed-position capture has the same single-return constraint when viewed from the scanner origin, since both formats represent what the scanner sees from one position. The constraint becomes meaningful only when comparing DepthPano to combined multi-setup point clouds, where the multi-setup combination can encode multiple returns through points captured from different positions.

For the Foundation's indoor accessibility-assessment context, single-return encoding is appropriate. The dimensional features that matter (doorway widths, clearances, riser heights, circulation envelopes) are bounded by the nearest reflective surface from the scanner position, and that surface is what the first or only return captures faithfully.

3.2 PNG Deflate compression is a modest contributor, not the headline

The intuition that spatially-coherent depth data should compress aggressively under PNG's Deflate algorithm is partially borne out and partially not. Deflate combines LZ77 dictionary coding with Huffman entropy encoding. Both perform best on data with repeating byte patterns. Depth values for a smooth surface change in small increments from pixel to pixel; the byte patterns at each step are similar but not identical, and the encoded depth values therefore do not produce the kind of exact byte-level repetition that LZ77 most efficiently compresses.

Differential encoding, storing each pixel's depth as a delta from its neighbour rather than as an absolute value, would produce much more compressible data for planar surfaces and is the approach taken by Kohira and Masuda (2017) for their PNG-based point cloud format. DepthPano does not currently use differential encoding; the absolute-value representation is simpler and integrates more cleanly with the broader bundle architecture, but at the cost of foregoing the compression advantage that differential encoding offers.

In operational use, DepthPano files compress to roughly 52–62% of their uncompressed size under PNG's Deflate, a reduction of approximately 38–48% on the bit-packed RGB byte stream, averaging 44.89% across the sample set (measured in §5.2.3). This is below the compression achievable by purpose-built point cloud compressors and below what differential PNG approaches would achieve on the same data. It is important to be clear that this 38–48% is the contribution of PNG Deflate alone, not the overall file-size reduction the architecture achieves. The end-to-end reduction relative to E57 is substantially larger (an approximately twenty-five-fold reduction in typical operational use, reported in Section 5) and comes mostly from the structural choices the architecture makes: removal of explicit azimuth and elevation coordinates, downsampling at image generation to the canonical resolution, separation of colour and intensity into the paired colour and intensity panoramas, and removal of per-point index information. PNG Deflate adds further compression on top of these structural savings, but it is one contributor among several and not the dominant one.

3.3 No native intensity or return classification

Standard point cloud formats can carry per-point return intensity, return number (first, second, last), and classification labels. DepthPano encodes only depth in the depth image; intensity is stored in the separate LaserPano layer (the reflected laser-return intensity) in the bundle architecture, and return classification is not stored at all because only one return is retained.

Applications that depend on return intensity for material discrimination can use the LaserPano layer; applications that depend on multi-return classification require an alternative format.

3.4 Projection-specific limitations

Equirectangular projection distorts heavily near the poles (top and bottom of the image). A scan position with the scanner placed on a tripod at typical indoor heights captures the ceiling near the top of the image and the floor near the bottom, both of which exhibit horizontal stretching as a consequence of the projection. The angular precision (radians per pixel) is uniform; the linear precision in pixels is not.

For the Foundation's accessibility-assessment context, the practical impact is limited because the features of interest are typically not at the poles. For applications focused on ceiling or floor geometry, alternative 2D projections (cube map, octahedral) may perform better. The Foundation chose equirectangular for integration with the broader bundle architecture, where all layers share the projection.

3.5 Standard versus hi-res: the operational trade-off

The choice between standard resolution (3142 × 1571) and hi-res (6284 × 3142) is an operational decision, not an architectural one. Both share the same projection convention and the same encoding; the only difference is the angular precision per pixel and, consequently, the capture time per scan position.

Hi-res scans take approximately twice as long to capture as standard scans, about ten minutes per scan position versus five, with the difference accounted for by the additional exposure and acquisition time the higher resolution requires. The device movement and re-levelling time between scan positions is roughly constant regardless of resolution. An average three-bedroom dwelling requires approximately fifteen scan positions to resolve the whole interior into floor plans and a 3D model. At standard resolution this is around 75 minutes of capture time across the dwelling; at hi-res it is around 150 minutes. The hour-and-a-quarter difference per dwelling is material at the scale of an operational therapist's caseload and material again at the scale of the sector.

We use standard resolution as the working default for the operational reasons described above. At two milliradians per pixel, approximately six millimetres of arc at three metres scan distance, the precision sits comfortably inside the NCC dimensional tolerances that matter for routine accessibility assessment. Clearances, gradients, riser heights, doorway widths, and circulation envelopes are all measured at scales for which six millimetres of angular precision is adequate.

Hi-res scans remain available as a configurable operator selection for contexts where the additional precision justifies the operational cost:

- Disputed scans where the exact dimensional value is contested and sub-NCC precision will be examined in detail.
- Tribunal-grade evidence baselines, where the cost of recapture is prohibitive and the scan must satisfy the highest level of scrutiny it might face.
- Pre-modification reference captures of properties where future modifications are anticipated and the baseline will be referenced repeatedly.
- Specific high-value locations within a dwelling; a contested doorway, a particular ramp; captured at hi-res while the rest of the dwelling is captured at standard.

The selective use case is supported architecturally by the calibration described in §2.1. Because the two resolutions share a projection convention, an operator can capture a dwelling at standard resolution and then re-capture specific locations at hi-res if a particular measurement is later contested. The hi-res scans overlay cleanly against the standard-resolution dwelling-scale model. The acquisition metadata records the resolution for each scan position, so a reviewer can determine at verification time whether a given measurement was derived from a standard or hi-res capture. This is particularly relevant in the selective-use case, where a dwelling-scale bundle may contain a mix of resolutions for different scan positions: each individually verifiable with its precision recorded.

4. Spatial queries

The operational benefit of DepthPano for the Foundation's workflow is not algorithmic complexity reduction. The implementation handles spatial queries on the CPU, and graphics acceleration is explicitly disabled in the browser viewer for performance reasons. The benefit is that spatial queries do not require Spherical-to-Cartesian conversion of the entire dataset before indexing. Direct array indexing by pixel coordinates retrieves the depth value at any direction in constant time.

4.1 Euclidean distance between two points

When a user selects two points in the DepthPano viewer, the application records their 2D pixel coordinates. The corresponding angular coordinates (θ, φ) are obtained from the pixel coordinates by inverting the equirectangular projection (Equations 1–2, with the azimuth-convention adjustment applied in Equations 5–6). The depth values r_1 and r_2 are retrieved by direct array lookup from the depth image.

The 3D Cartesian coordinates of each selected point relative to the scanner origin are:

$$X = r \cdot cos(\varphi) \cdot cos(-\theta) \quad (5)$$

$$Y = r \cdot cos(\varphi) \cdot sin(-\theta) \quad (6)$$

$$Z = r \cdot sin(\varphi) \quad (7)$$

Note the inversion of the azimuth, which matches the mathematical convention for trigonometry. The Euclidean distance D between the two points is then:

$$D = \sqrt{((X_2 - X_1)^2 + (Y_2 - Y_1)^2 + (Z_2 - Z_1)^2)} \quad (8)$$

The operation completes within a few milliseconds in the browser, dominated by the JavaScript overhead rather than the arithmetic. The advantage over a point cloud-based implementation is that no preliminary indexing step is required: the depth values are addressable directly by pixel coordinates without building or maintaining a spatial index.

4.2 Angle between two directions

To measure the angle between two surfaces or features (for example, verifying that wall intersections are orthogonal, or measuring the gradient of a ramp), the Cartesian coordinates produced by the distance computation can be repurposed. With v_1 and v_2 as the vectors from scanner origin to the two selected points, the included angle α is:

$$\alpha = \arccos((v_1 \cdot v_2) \,/\, (|v_1| \cdot |v_2|)) \quad (9)$$

For accessibility-assessment applications, the most common angular queries are gradient measurements (ramp slope, threshold rise/run) and orthogonality verification (wall intersections, floor-wall right angles). The DepthPano structure supports these queries without requiring specialist tools.

4.3 Operational characteristics

The query operations run in the client browser without graphics acceleration. The Foundation's reference implementation explicitly disables WebGL acceleration in the browser viewer, on the basis that the data retrieval is the limiting factor and the arithmetic is straightforward enough that CPU execution is competitive with GPU execution for the query sizes involved (typically one to a few measurements per user interaction). The implementation choice is operational rather than principled: GPU-accelerated implementations of the same queries are possible and may be preferred in future versions, but the current implementation does not use them.

A small consequence of the calibration in §2.1 worth noting at this point: because the conversion from azimuth and elevation to pixel coordinates is multiplication by an exact decimal (×1000 at hi-res, ×500 at standard), the array indexing expression for depth lookup is essentially Array[θ × 1000, φ × 1000]. This is unusually direct for spatial query code; readers of the reference implementation routinely find the indexing legible without commentary, which reduces the maintenance burden on the codebase and makes independent verification more tractable.

5. Test results

This section reports file-size and query-performance results on a sample scanned domestic residence from the Foundation's operational scan corpus. The test case has 15 scans.

In addition, the summary of results shows a statistical summary of the same calculations for the Foundation’s corpus which consists of 345 BLK360 scans from 25 residential dwellings, captured in ACT, Victoria and New South Wales.

All scans are at the Foundation's standard resolution (3142 × 1571) unless otherwise noted.

5.1 Test scan characteristics

The 15 scans are for a variety of dwelling spaces. It is important to consider the geometric characteristics of these spaces in particular when analysing geometric accuracy of the DepthPano. Smaller rooms or other internal spaces will have better accuracy than external spaces or large, open plan areas with large windows or French doors.

Table 1: Scan space classification

Type of space	Scans	Distances	Maximum distance
Foyer with glass front overlooking external garden and trees.	#1, #11	Small + Medium + Large	50m
Corridor	#2, #9, #10	Small + Medium	30m
Internal spaces with few or no external windows	#3, #6, #7, #8	Small	5m
Open plan room	#4, #5, #13	Medium	30m
Open plan room + French doors	#12, #15	Medium + Large	50m
Car port (external space enclosed on two sides)	#14	Small + Large	50m

For the purposes of this classification a small distance is one up to 5 metres, medium is from 5 to 30 metres and large is 30 metres and above. Note, this is only an approximate characterisation and some outliers may exist within each scan. More detailed information for the scans is included in [Table 2]

Table 2: Distance statistics for the scans

Scan	Average distance (m)	Points within 5m (%)	Maximum distance (m)
1	2.26	97.22	54.56
2	1.57	97.28	48.18
3	1.03	100.00	4.78
4	1.75	98.04	37.55
5	2.42	94.61	47.02
6	2.04	97.41	23.40
7	1.57	98.52	29.07
8	1.39	100.00	4.58
9	1.74	96.76	39.12
10	2.03	98.38	54.10
11	2.40	97.20	53.81
12	2.36	94.82	36.78
13	1.93	96.94	37.55
14	2.40	91.09	29.32
15	2.51	91.95	35.00

Things to note from this information:

- Scans #3 and #8 have all points within 5m of the scanner. They are enclosed small rooms with no windows.
- Larger distances are for external features (e.g. trees) which appear in the scans due to spaces including windows or glass doors.

5.2 Decomposition of file-size savings

Several factors contribute to the file-size savings. These include:

- Downsampling
- Dimensions per sample
- Compression

These factors and how they are handled in the test results are explained further in the following sections.

5.2.1 Downsampling

A DepthPano standard resolution represents a total of 4,936,082 interpolated samples in a 3142 x 1571 grid.

E57 point clouds used in the study range in size from 11,596,928 samples (4816 x 2408 grid) to 12,103,200 samples (a 4920 x 2460 grid). The DepthPano standard resolution therefore represents approximately 40.8% to 42.6% of the samples of the corresponding E57 point cloud.

5.2.2 Dimensions per sample

Each E57 sample provides spherical coordinates (azimuth, elevation and range), RGB colour values, and intensity values. In addition azimuth and elevation indices are also provided.

The DepthPano samples only contain spherical coordinate range information. The azimuth and elevation are not stored since they are implicit based on the coordinates of the pixel being examined. The test examples also include RGB colour values stored in a corresponding standard PNG file which provides a significant saving for this information. The intensity information is provided as a separate LaserPano PNG file.

Note, since the data dimensions of RGB and intensity values are fully expressed by the test outputs the cost saving is primarily based on the removal of the azimuth and elevation values and the grid indices which are provided by the PNG's two-dimensional structure.

5.2.3 Compression

E57 files are not significantly compressed. A significant saving can be made by compressing the raw E57 and uncompressing it before use. [Table 3] compares the file sizes of the raw E57 files with a compressed version of the file using the LZMA (Lempel–Ziv–Markov chain algorithm) compression algorithm for the sample project.

Table 3: E57 compression ratios

Scan	Raw (kB)	Compressed (kB)	File size ratio (%)	Saving (%)
1	232,930	73,957	31.75	68.25
2	229,916	75,051	32.64	67.36
3	230,865	74,687	32.35	67.65
4	233,168	74,219	31.83	68.17
5	231,583	75,247	32.49	67.51
6	232,211	74,773	32.20	67.80
7	231,780	73,311	31.63	68.37
8	232,211	75,588	32.55	67.45
9	230,680	74,870	32.46	67.54
10	228,963	74,748	32.65	67.35
11	238,954	74,051	30.99	69.01
12	232,544	76,656	32.96	67.04
13	231,631	70,241	30.32	69.68
14	230,632	74,944	32.50	67.50
15	231,780	74,459	32.12	67.88
Average	231989.87	74453.47	32.10	67.90

E57 format files are highly suitable for compression. They include sentinel values for missing readings and index fields which have a clearly defined progression. The values recorded confirm this with an average saving of over 68%.

The DepthPano uses the standard PNG data compression based on the LZ77 algorithm. [Table 4] gives the file size ratio of the compressed file when compared to an uncompressed raw file.

Table 4: DepthPano compression ratios

Scan	Raw (kB)	Compressed (kB)	File size ratio (%)	Saving (%)
1	14,464	7,571	52.34	47.66
2	14,464	7,788	53.84	46.16
3	14,464	8,972	62.03	37.97
4	14,464	7,822	54.08	45.92
5	14,464	7,773	53.74	46.26
6	14,464	7,936	54.87	45.13
7	14,464	7,923	54.78	45.22
8	14,464	8,696	60.12	39.88
9	14,464	7,807	53.98	46.02
10	14,464	7,853	54.29	45.71
11	14,464	7,504	51.88	48.12
12	14,464	7,942	54.91	45.09
13	14,464	7,650	52.89	47.11
14	14,464	8,263	57.13	42.87
15	14,464	8,061	55.73	44.27
Average	14,464	7970.73	55.11	44.89

Compression for the PNG DepthPano files is reasonably modest with an average saving of 44.89% across the sample set (individual scans range from 37.97% to 48.12%). The primary component of this compression is the encoding of sentinel values for missed readings. The scans have the bottom 30 degrees, 16.6% of the total scan, filled with missed readings as this is the limit of the scanner. This data is then compressed to a single value for each 258 bytes of the run-length pattern.

5.3 File-size comparison against E57

A typical residence with 15 individual scans was used for this test. Comparisons were made between a compressed version of the source E57 file for the scan and the corresponding DepthPano total bundle size including colour and intensity files. All file sizes are given in kB.

Table 5: File size comparison of DepthPano against E57

Scan	Source E57		DepthPano				File size ratio (%)	Saving per sample (%)
	Raw	Compressed	Depth	RGB	Intensity	Total		
1	232,930	73,957	7,439	215	398	8,052	10.89	36.54
2	229,916	75,050	7,676	182	440	8,298	11.06	36.47
3	230,865	74,687	9,091	180	494	9,765	13.08	35.64
4	233,168	74,219	7,693	196	469	8,358	11.26	36.38
5	231,583	75,247	7,673	240	417	8,330	11.07	36.46
6	232,211	74,772	8,084	216	450	8,750	11.70	36.20
7	231,780	73,311	7,971	201	466	8,638	11.78	36.17
8	232,211	75,587	9,140	230	410	9,780	12.94	35.69
9	230,680	74,870	7,830	192	452	8,474	11.32	36.36
10	228,963	74,747	7,754	188	400	8,343	11.16	36.42
11	238,954	74,051	7,395	208	408	8,011	10.82	36.56
12	232,544	76,656	7,823	250	433	8,506	11.10	36.45
13	231,631	70,241	7,537	212	425	8,174	11.64	36.23
14	230,632	74,943	8,048	378	446	8,872	11.84	36.15
15	231,780	74,459	7,946	251	418	8,614	11.57	36.26

Key:

- File size ratio represents the size of the DepthPano bundle compared to compressed E57 file.
- Saving per sample incorporates the downscaling factor (41% to 42%) in file size factor to give an approximate saving per sample between the compressed E57 file and the corresponding DepthPano bundle.

5.4 Summary of file size savings

[Table 6] gives the average file sizes (kB) for the 345 scans of the Foundation's corpus and the compression savings.

Table 6: Summary of compression savings

Measure	Source E57		DepthPano				File size ratio (%)	Saving per sample (%)
	Raw	Compressed	Depth	RGB	Intensity	Total		
Average	234,486	76,532	8,097	296	490	8,883	11.61	36.24
Minimum	231,583	73,745	9,192	273	570	10,034	13.61	35.42
Maximum	239,680	77,347	7,080	271	445	7,796	10.08	36.87

Note, the Minimum and Maximum rows report the scans with the minimum and maximum number of measured points respectively, not the minimum and maximum of each column. In this corpus a lower measured-point count is associated with a higher file-size ratio (a larger DepthPano bundle relative to its compressed E57), and a higher point count with a lower ratio; this is why the Minimum row carries the larger figures and the Maximum row the smaller. A decomposition of the file savings for each component of the DepthPano conversion process is shown in [Table 7]. Note, whilst E57 compression is not strictly part of the conversion process it is included here as it would be possible to convert the uncompressed E57 to a compressed format which would give similar savings.

Table 7: Decomposition of file saving by component of conversion

Component of work flow	Average file size (kB)	File size saving (%)	File ratio for component (%)	Running file ratio (%)
E57 compress	76,532	67.4	32.6	32.6
Downsampling	31,378	59.0	41.0	13.4
Implicit coordinates	15,250	51.4	48.6	6.5
PNG compress	8,883	42.0	58.0	3.8

5.5 Performance

Overview

A number of factors must be considered when comparing the performance of the DepthPano image versus the original E57 files:

- Time to load the data file
- Time to build the model (E57 only)
- Time to access the model / image to retrieve a depth value for a corresponding azimuth and elevation.

Note, the E57 data requires that the data is stored in a model allowing easy access. In the general case point cloud data is stored in an octree. However, for the comparison to the DepthPano the E57 data is stored in a quadtree with the two dimensions of azimuth and elevation. This allows a nearest neighbour function to be called out for specific azimuth and elevation values.

Hardware details

CPU: AMD Ryzen 7800X3D 8-core, 4.2 GHz clock speed

Memory: 32 GB, 66 GB/s

Disk drive: 2 TB SSD, read 300 MB/s, write 330 MB/s (approximately)

Results: Model load timings

[Table 8] gives the times for the loading of the files for the two models DepthPano and E57. The E57 time includes the construction of a quadtree to allow the model's use for nearest neighbour operations.

Table 8: Comparison of model load times (ms)

	Model load times, including build model for E57 (ms)		
	Average	Minimum	Maximum
E57	2,677	2,656	2,690
DepthPano	140	132	149

The majority of the time for the loading of the E57 is taken building the quadtree for future processing. The tree is built as the data is loaded and therefore the component of the time spent actually loading the data cannot be measured. However, given the disk speed and average file size approximately 700ms is spent reading the data and almost 2 seconds (1.923s) building the tree.

Results: Get depth value timings

[Table 9] gives the relative times for retrieving depth data from the models.

Table 9: Comparison of times for depth retrieval (µs)

	Times for depth retrieval (µs)		
	Average	Minimum	Maximum
E57	4.6	2.2	22.5
DepthPano	1.0	0.7	1.4

Whilst the access time for the E57 data is over 4 times that for the DepthPano it is unlikely to have a major impact on performance for typical operations (e.g. measuring the distance between two points). It will however have more impact for operations requiring the reading of all of the scan data. An example of the latter would be geometric feature extraction (surfaces and other planes). The exact impact of this is left for future research.

5.6 Geometric accuracy

The application CloudCompare was used to compare the original E57 point cloud with the point cloud represented by the DepthPano image. The DepthPano image was first converted to a Point Cloud Library Data file (PCD) to allow import into CloudCompare. The DepthPano point cloud was first registered against the E57 point cloud before the distances (Point Cloud to Point Cloud) were calculated. The Iterative Closest Point (ICP) algorithm was used for the registration with default parameters: $1e^{-6}$ minimum error with 100% model overlap. A summary of the statistical values for the error distances of the scans under test is included in [Table 10].

Table 10: Summary of distance errors (mm)

Scan	Distance variation (mm)						Average distance (m)	Readings < 5m (%)
	Readings	Mean	Median	Standard deviation	95 percentile	99 percentile		
1	3,823,981	6.136	4.173	7.741	17.385	30.964	2.264	97.22
2	3,950,589	3.473	1.997	5.66	10.557	25.50	1.572	97.27
3	3,948,035	1.042	0.858	0.777	2.393	4.161	1.03	100.00
4	3,900,176	1.251	0.937	1.229	3.166	6.468	1.747	98.04
5	3,944,609	2.56	2.044	2.111	6.218	10.637	2.418	94.61
6	3,951,648	1.744	1.417	1.397	3.982	7.25	2.036	97.41
7	3,948,868	2.009	1.357	2.133	5.034	10.402	1.569	98.53
8	3,932,720	0.992	0.773	0.799	2.425	3.904	1.39	100.00
9	3,928,308	2.701	1.905	3.11	6.926	14.571	1.74	96.76
10	3,955,554	2.211	1.79	2.089	4.74	9.493	2.033	98.38
11	3,784,057	6.394	4.488	7.328	17.728	30.773	2.399	97.22
12	3,889,388	2.383	1.795	2.235	6.202	11.298	2.361	94.83
13	3,839,634	2.209	1.493	2.595	5.874	14.051	1.934	96.93
14	3,805,430	2.687	1.745	2.696	7.715	13.401	2.395	91.09
15	3,924,937	1.773	1.246	1.882	5.094	9.364	2.507	91.95

The distances vary significantly for different types of scans:

- Scans 1 and 11 are of a glass fronted foyer which overlooks an external space which also leads to higher distances and correspondingly higher error values.
- Scans 3 and 8 are for rooms with no external windows and therefore give better overall errors and all distances are within 5m of the scan sensor.

A histogram with the distribution of distance variations for the scans is shown in [Figure 1].

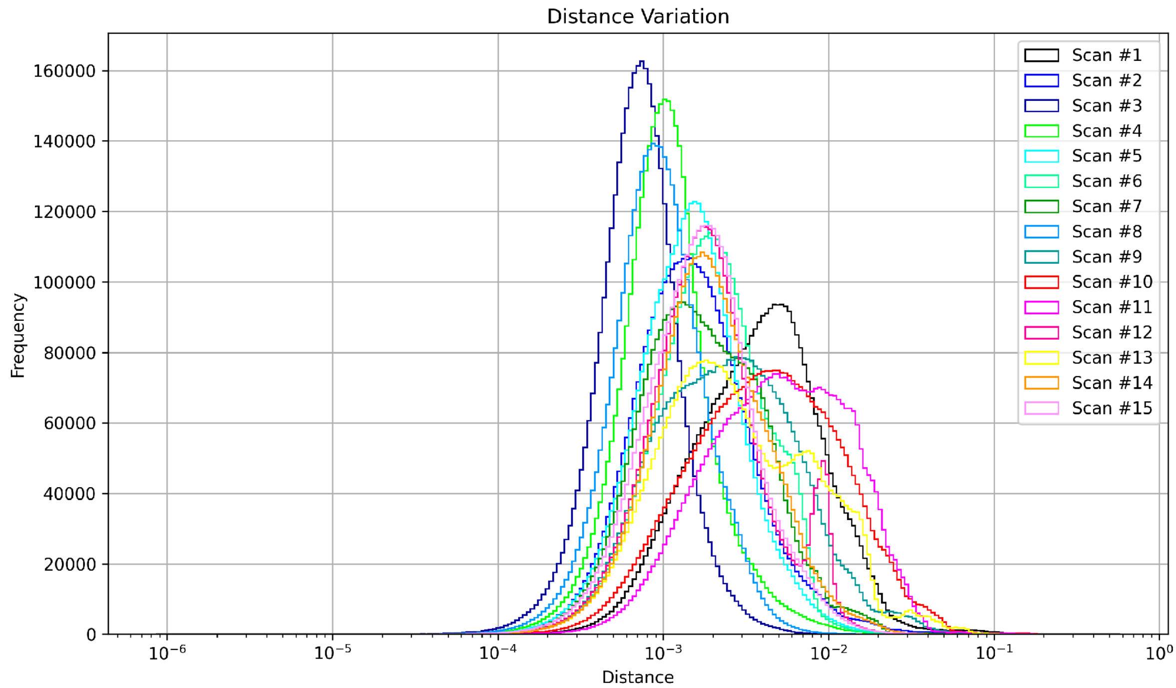


Figure 1: Distance error variation for standard DepthPano (metres)

Note, there is considerable variation between the scans. For interior spaces the mean error values are between 1mm and 3mm. For spaces with large exterior windows the mean values are closer to 1cm.

[Figure 2] shows the typical distribution of these distance errors, specifically for scan #2. Far objects including the external tree have significantly higher errors than the nearer surfaces where the errors tend towards the mean error.

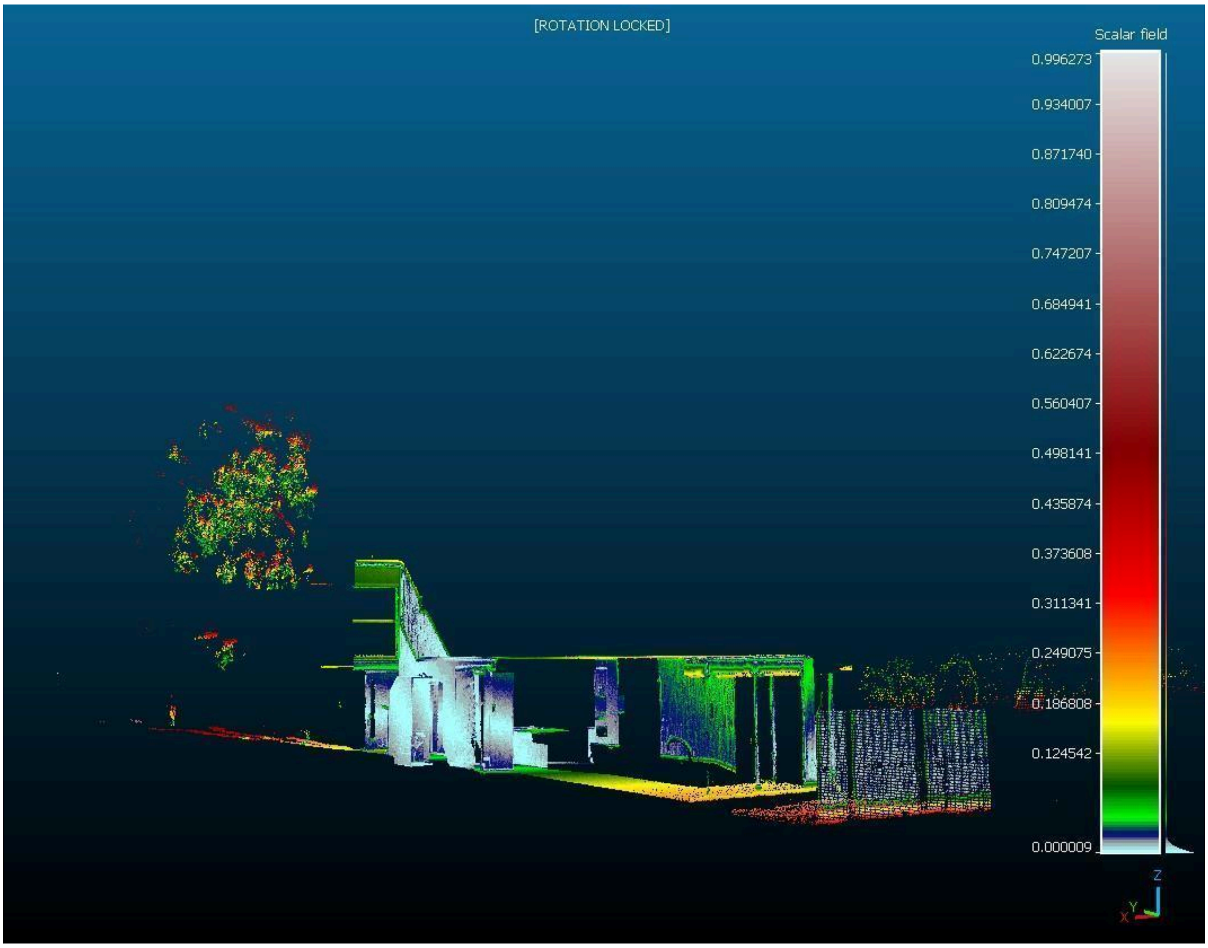


Figure 2: Error heat-map for scan #2

5.7 Performance of the hi-res DepthPano

The experiment was repeated with hi-res DepthPanos (6284 x 3142 pixels). There is no downscaling with the hi-res DepthPano. It contains approximately 30% more points than the original E57 scan, so the saving per sample is not a meaningful value.

Table 11: Compression for high resolution DepthPano

Measure	Source E57		DepthPano				File size ratio (%)	Saving (%)
	Raw	Compressed	Depth	RGB	Intensity	Total		
Average	232,307	72,798	30,309	911	913	32134	44.14	55.86
Minimum	232,930	73,957	57,852	796	800	57052	77.14	22.86
Maximum	238,132	59,084	20,155	1614	1617	23386	39.58	60.42

As in Table 6, the Minimum and Maximum rows report the scans with the fewest and most measured points respectively, not the minimum and maximum of each column; the scan with the fewest points carries the higher file-size ratio and the larger totals, and the scan with the most points the lower. [Table 12] shows the distance errors for the high resolution compared to the original E57 scans.

Table 12: Distance errors for high resolution DepthPano (mm)

Scan	Number of points	Distance variation (mm) Mean	Median	Standard deviation	95 percentile	99 percentile	Average distance (m)	Readings < 5m (%)
1	15,295,153	4.01	2.70	4.96	11.07	20.04	2.26	97.22
2	15,799,148	3.33	1.82	5.59	10.45	25.57	1.57	97.28
3	15,791,951	0.88	0.73	0.63	1.91	3.35	1.03	100.00
4	15,597,221	2.12	1.60	2.17	5.32	11.57	1.75	98.04
5	15,776,012	2.36	1.68	2.18	7.55	9.61	2.42	94.62
6	15,805,314	3.65	2.69	3.59	9.76	17.52	2.04	97.41
7	15,794,810	2.39	1.62	2.35	6.64	10.91	1.57	98.54
8	15,731,736	0.89	0.67	0.78	2.29	3.78	1.39	100.00
9	15,712,808	4.07	2.40	5.87	12.00	28.73	1.74	96.76
10	15,821,552	1.77	1.31	1.94	4.39	8.92	2.03	98.38
11	15,135,485	3.92	2.87	4.22	9.89	17.45	2.40	97.22
12	15,557,315	3.24	2.38	3.21	8.83	15.93	2.36	94.83
13	15,358,077	3.90	2.30	5.13	11.54	27.08	1.93	96.94
14	15,222,043	2.55	1.92	2.38	6.84	12.49	2.40	91.08
15	15,700,136	3.96	2.81	3.96	11.19	19.39	2.51	91.96

Rather than examining the individual values of this table it is easier to compare these values with those of the low-resolution values to see if the errors have improved. The comparison as a percentage of the low-resolution DepthPano is included in [Table 13]. Values over 100.0% indicate a worse value for the hi-resolution DepthPano than the low-resolution DepthPano.

Table 13: Ratio of errors for hi-res versus standard DepthPano

Scan	Mean	Median	Standard deviation	95 percentile	99 percentile
1	106.12	100.08	103.33	115.14	109.50
2	111.89	104.39	128.59	116.73	122.92
3	112.53	111.69	114.24	112.58	116.47
4	130.07	129.96	124.77	118.73	131.72
5	93.50	92.76	95.27	94.05	94.64
6	107.19	105.17	130.14	102.05	150.37
7	92.22	98.10	77.22	86.59	65.37
8	88.79	85.61	94.16	91.36	93.30
9	62.18	67.05	56.24	56.63	55.45
10	48.80	60.31	39.21	39.01	37.92
11	70.80	65.13	80.43	74.88	85.98
12	246.85	290.82	209.74	182.85	214.27
13	69.64	97.72	54.15	56.58	54.50
14	168.65	166.31	186.74	184.75	183.23
15	171.66	167.66	178.71	174.39	184.60

8 of the 15 hi-res DepthPanos had worse errors than the corresponding low-resolution DepthPanos whereas 7 had better results. These are highlighted in red. It is difficult to assess the exact reasons of this variation particularly on scans in the same spaces. For example: scan #1 and scan #11 had very different results though they were taken in the same space from a similar position. It is likely that the oversampling is causing an exaggeration of errors in different areas.

A histogram for the errors in each of the scans is shown in [Figure 3]. In general the extra points lead to a smoothing of the peaks of the graph from the standard DepthPanos. Whilst the peaks tend to be at higher values the shapes are not significantly different.
We therefore recommend that hi-res DepthPanos are not used as the default for standard scans, as the improvement of geometric accuracy cannot be guaranteed and the files produced are much larger (over 3 times larger). This recommendation concerns routine, dwelling-wide capture and does not contradict the selective hi-res use cases set out in §3.5: that section concerns situations where additional angular precision is required for specific contested or high-value measurements, whereas the finding here is that hi-res does not reliably improve point-to-point fidelity against the E57 reference across general scans, so it should not be adopted wholesale.

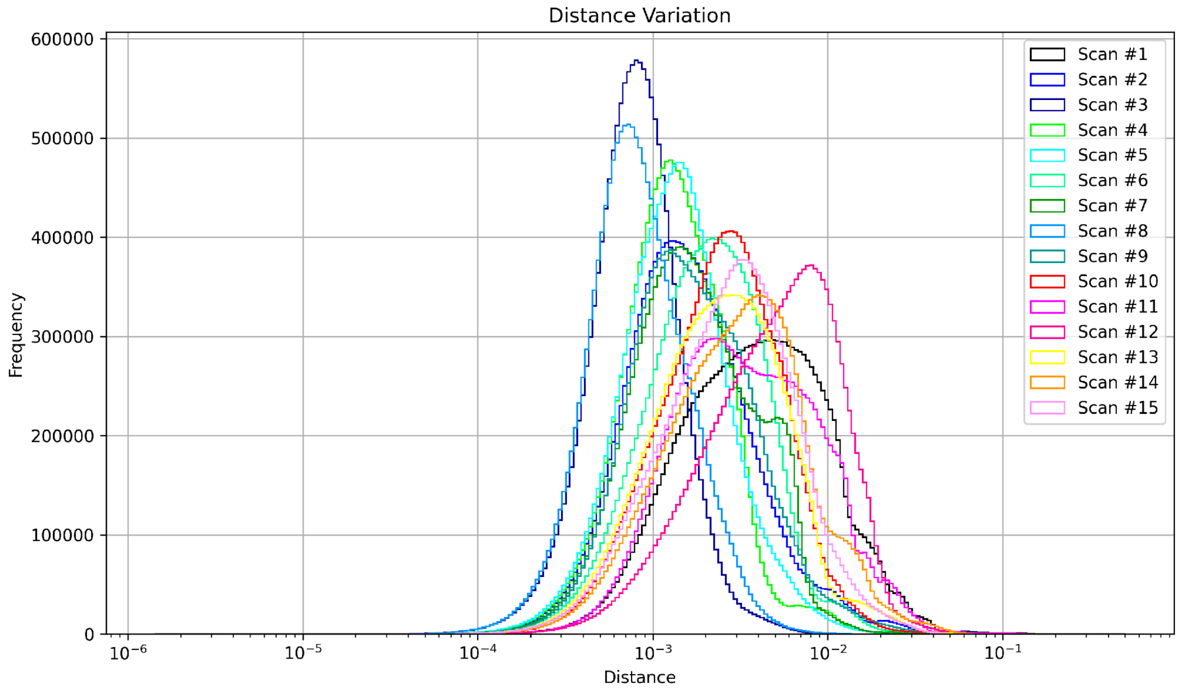


Figure 3: Distance error variation for hi-res DepthPano (metres)

6. Comparison with prior work

Three published works are directly relevant to DepthPano and warrant explicit comparison. The Foundation's position is honest about what each does better and what DepthPano does differently for its operational context.

6.1 MPEG V-PCC

The MPEG V-PCC codec uses 2D mapping as the basis for point cloud video compression. V-PCC achieves substantially better compression than DepthPano for the use cases it targets (animated point cloud data, temporal coherence between frames). V-PCC is the appropriate reference for the broad architectural family within which DepthPano sits.

DepthPano differs from V-PCC in three ways relevant to the Foundation's use case. First, V-PCC is designed for animated content with temporal coherence; DepthPano is designed for static captures and does not exploit temporal compression. Second, V-PCC is implemented through proprietary or licensed codec stacks; DepthPano uses PNG, which has open and ubiquitous decoder support. Third, V-PCC's complexity is appropriate for high-throughput streaming applications; DepthPano's simpler structure is appropriate for the modest-throughput review workflows the Foundation supports.

6.2 Kohira and Masuda (2017): differential PNG

Kohira and Masuda demonstrated that PNG-based 2D representation of point cloud data achieves competitive compression when the data is encoded differentially: each point's coordinates expressed as a delta from a reference rather than as an absolute value. For mobile mapping data with smooth, planar surfaces, this approach achieves substantially better compression than absolute-value encoding.

DepthPano uses absolute-value encoding, not differential encoding. The Foundation's reasons for this choice are practical: absolute-value encoding is simpler to decode in the browser viewer, simpler to integrate with the rest of the bundle architecture, and simpler to verify cryptographically (each pixel's value is a self-contained measurement rather than depending on a chain of differences from a reference).

The trade-off is real. DepthPano gives up the compression advantage of differential encoding for the operational and architectural simplicity of absolute-value encoding. Adopters whose primary requirement is file-size minimisation should consider Arikawa's approach as an alternative.

6.3 Dumić et al. (2022): equirectangular evaluation

Dumić et al. evaluated several 2D projections for point cloud compression and found that equirectangular projection was not the strongest performer in their tests, both for compression ratio and for point-to-point geometric accuracy. Cube-map and octahedral projections performed better on their benchmark.

The Foundation's choice of equirectangular projection for DepthPano is therefore not driven by its independent geometric advantages. The choice is driven by integration with the rest of the bundle architecture: the ColourPano, the LaserPano (reflected intensity), the optional HeatPano (thermal), the NormalPano (surface-normal) layer, and the architectural-interpretation layers all use the same equirectangular projection, and the cryptographic verification depends on co-registration of all layers at the pixel level. Switching to a non-equirectangular projection for DepthPano alone would break this co-registration; switching the entire bundle to a different projection is an architectural decision with consequences far beyond compression performance.

Adopters whose architecture does not have the same co-registration requirements should consider Dumić's findings when choosing a projection. For the Foundation's bundle architecture, the choice is constrained by integration requirements rather than by independent merit.

7. Conclusion

DepthPano is the depth-image format the Metrological Research Foundation uses for indoor terrestrial laser scan data in residential and small-commercial settings. The format encodes radial depth as a 24-bit fixed-point value across the RGB channels of a PNG image at equirectangular projection, with a per-image scaling factor stored in a tEXt chunk. The format is one application of an established family of 2D-projection techniques for point cloud data; the contribution of this paper is the documented implementation and its operational results, not a novel architectural idea.

End-to-end file-size reductions of approximately twenty-five-fold relative to E57 are achieved in operational use: a typical 230 MB E57 produces a DepthPano bundle of roughly 8.5 to 9.5 MB. The reduction is attributable mainly to the structural choices the architecture makes (removal of explicit coordinate data, downsampling to the canonical resolution, separation of attributes into purpose-specific files) rather than to compression of the depth values per se; PNG's Deflate compression contributes a more modest 38–48% on the bit-packed depth bytes, on top of these structural savings. Geometric accuracy is preserved within the operational tolerances required for NCC dimensional compliance verification. Spatial queries are supported through direct array indexing, avoiding the Spherical-to-Cartesian conversion that point cloud formats require for equivalent operations.

Comparison with published prior work shows that other approaches achieve better compression (Arikawa's differential PNG) or better geometric accuracy (Dumić's cube-map projection) on the dimensions evaluated in their respective benchmarks. The Foundation's choice of DepthPano is driven by integration with the rest of the bundle architecture for verifiable spatial evidence, not by independent compression or accuracy advantages. Adopters with different architectural requirements should consider the alternatives identified here.

The format is published in the public domain under open access terms. The Foundation has no patent claims on the format itself. The paper is offered as documentation, as a basis for further engineering work in the field, and as a transparent description of one application of 2D-projection point cloud techniques to a specific operational context.

Appendix A. Image generation pipeline

A.1 Overview

The algorithm uses a scan specified using spherical coordinates. If the scan is presented in Cartesian coordinates, it must be first translated to spherical coordinates. The DepthPano creation algorithm takes advantage of an input scan sorted into an evenly spaced grid of monotonic rows (azimuth) and columns (elevation). The dimensions of the grid should match the output of the process as closely as possible. The generation algorithm is optimised to take advantage of this grid. Each pixel in the output image is interpolated from the depths of the quadrilateral of the grid containing the pixel's azimuth and elevation. Note, initially the depth values are stored in floating point format, but they are normalised using the maximum depth of the interpolated scan to scale the depths to fit into 24-bit RGB values.

A.2 Sorting Input Data

The algorithm requires that the input data consists of a two-dimensional array of monotonic rows and columns where the rows are monotonic with respect to azimuth and the columns with respect to elevation. This optional step sorts the source data into a two-dimensional grid based on azimuth and elevation. The rows and columns must be monotonic with respect to their dimension. If array is not monotonic for each row and column it must be sorted. Sort the data into a 2x1 grid with twice as many columns as rows.

```
SORT all points by azimuth (use elevation when azimuth of two points equal)
DIVIDE data into groups of size n (n = number of points per column)
FOR EACH column
  SORT points in row by elevation
END FOR
```

Pad the grid on the bottom edge with missing reading values with the same coordinates as the existing lowest point for that column.

Note, most scans are already sorted into this format so check the arrangement of the data to see if sorting is required.

A.3 Main Algorithm Loop

The following pseudo code describes how the scan grid is traversed and the distance array is created. Note, when interpolating each point a quadrilateral from the scan is considered using the points P_{00} P_{01} P_{10} P_{11} where the indices show the azimuth and elevation of the quadrilateral (zero for the value less than the point being calculated and 1 representing the value higher than the point being calculated). The current point refers to a pixel reference into the output grid and has an associated azimuth and elevation for that pixel.

```
FOR each column
  SET current point to column azimuth and elevation -90 degrees
  SET current quad using bottom row of scan to contain current point azimuth
  WHILE NOT current point elevation in starting quad
    set current point distance to null value
    increment current point elevation to next pixel
  END WHILE
  WHILE NOT column done
    check top edge of current quad (P01 and P11)
    IF current azimuth between the left and right point of the top edge
      SET intersect line to between top points (P01 P11)
```

```
    SET next quad to above current quad
  ELSE
    IF current quad P01 (left top) azimuth > current point azimuth
             SET intersect line to left of current quad (P00 P01)
             SET next quad to left of current quad
    ELSE (right point must be lower)
             SET intersect line to right edge of current quad (P10 P11)
             SET next quad to right of current quad
    END IF
  find intersect point on intersect line for current point azimuth
  IF intersect point elevation > +90 degrees
           set the intersection point to 90 degrees
  END IF
  WHILE intersection point elevation is > current point
    SET current point distance based on interpolation of quadrilateral
    INCREMENT current point elevation
    IF current point elevation >= +90 degrees
      SET column done to true
      BREAK
    END IF
  END WHILE
  IF not column done
    SET quad to next quad
  END IF
 END WHILE
END FOR
```

Note, the azimuth values may wrap (e.g. a value of 5 degrees is considered as being > 355 degrees). Also, for the elevation at the top of the scan the final quad may also wrap by considering the quadrilateral 180 degrees away from the existing scan line below the previous point.

A.4 Interpolation of each point

The interpolation for each point in the output image could use bilinear interpolation via a mapped square. This is costly (non-linear) but provides a smoother response. However, for efficiency the quad is split into triangles and then barycentric interpolation is used. This is simple, fast, and numerically stable.

Note, interpolation must only be used for points which are coplanar. For points that are not coplanar, the nearest-neighbour value must be used. The simple method to determine this is to check that the distance values for the three points of the triangle used fall within a similar range. By experimentation a value of 0.1m for the distance differences was chosen as an assumption that the points are co-planar. Even if the points are not on the same plane the use of this value means that edges of openings (e.g. doors) into other spaces will not be interpolated leading to invalid values existing between the two surfaces.

To choose which triangles to decompose the quad into the "twist" method is used. This provides a decomposition with the least distortion.

Define edge vectors:

$e0 = P_{10} - P_{00}$
$e1 = P_{11} - P_{01}$
$e2 = P_{01} - P_{00}$
$e3 = P_{11} - P_{10}$

Compute twist:

T=(e0×e1)·(e2×e3)

The twist then determines how to decompose the quad:

```
IF twist == 0
  // all points coplanar
  // doesn't matter how the quad is decomposed.
  use diagonal P00 P11
ELSE
  IF twist > 0
    use diagonal P00 P11
  ELSE
    // twist < 0
    use diagonal P01 P10
  END IF
END IF
```

Note, for the quadrilaterals being used the cross products collapse (azimuth and elevation values cancel out) so that the twist calculation can simply use the range values for the four points of the quadrilateral.

Pseudo code for the interpolation including the twist determination follows. The parameter quad has the four points p00, p01, p10 and p11. Each point has the values: x = azimuth, y = elevation, z = range.

```
FUNCTION barycentric_weights(p, a, b, c):
    denom = (b.y – c.y)*(a.x – c.x) + (c.x – b.x)*(a.y – c.y)

    w1 = ((b.y – c.y)*(p.x – c.x) + (c.x – b.x)*(p.y – c.y)) / denom
    w2 = ((c.y – a.y)*(p.x – c.x) + (a.x – c.x)*(p.y – c.y)) / denom
    w3 = 1 - w1 - w2

    RETURN w1, w2, w3
END FUNCTION

FUNCTION point_in_triangle(w1, w2, w3, eps=1e-9)
    RETURN (w1 >= -eps) AND (w2 >= -eps) AND (w3 >= -eps)
END FUNCTION

FUNCTION interpolate_quad_adaptive(point, quad)
    # Compute twist
    twist = quad.p00.z + quad.p11.z – quad.p10.z – quad.p01.z

    IF twist >= 0
        # Use diagonal P00 to P11
        w1, w2, w3 = barycentric_weights(p, quad.p00, quad.p10, quad.p11)
        IF point_in_triangle(w1, w2, w3)
            # Triangle A
            RETURN w1*z00 + w2*z10 + w3*z11
        ELSE
            # Triangle B
            w1, w2, w3 = barycentric_weights(p, quad.p00, quad.p11, quad.p01)
            RETURN w1*z00 + w2*z11 + w3*z01
        END IF
```

```
  ELSE
    # Use diagonal P10 to P01
    w1, w2, w3 = barycentric_weights(p, quad.p00, quad.p10, quad.p01)
    IF point_in_triangle(w1, w2, w3)
      # Triangle A
      RETURN w1*quad.p00.z + w2*quad.p10.z + w3*quad.p01.z
    ELSE
      # Triangle B
      w1, w2, w3 = barycentric_weights(p, quad.p10, quad.p11, quad.p01)
      RETURN w1*quad.p10.z + w2*quad.p11.z + w3*quad.p01.z
    END IF
  END IF
END FUNCTION
```

A.5 Scaling

The floating point distances are stored in an array when creating the image initially from the scan. The distances array is the same dimensions as the output image. The maximum distance is stored as the array is populated. When the distance array is complete a scaling factor is calculated using the maximum distance and the maximum value that can be stored in the colour 24-bits of the image. The maximum value storable is FFFFFF hex or 16,777,215 decimal. An image is then created with each pixel equal to the distance array value multiplied by the scaling factor. The scaling factor is stored in the output PNG image in a tEXt field with key “DepthPano:scale”.

A.6 Handling missing values

Not all points have valid readings. If points are missing from the quad currently being filled the following procedure is used.

1 value missing

If the point being processed lies within the triangle of valid points the value is interpolated. Otherwise, it is unset.

2 values missing

If the point being processed is close to one of the two valid points it is set to an interpolated value based on the two points. Otherwise it is unset. Close is defined as being within a quadrant on the quadrilateral of one of the two points.

3 values missing

If the valid point is close to the point being processed it is set to its value. Otherwise it is unset. Close is defined as being within a quadrant of the point within the quadrilateral.

4 values missing

If all depth values of the quadrilateral are unset the output point is also unset.

Appendix B. Encoding specification

This appendix specifies the DepthPano file format precisely, sufficient for an independent implementer to produce conformant files or to decode files produced by the Foundation's implementation.

B.1 File format

DepthPano files are PNG files conforming to the W3C PNG specification (ISO/IEC 15948). The IHDR chunk specifies an 8-bit RGB image (Colour Type 2, Bit Depth 8). The Alpha channel is not used; files with an Alpha channel (Colour Type 6) are tolerated by conformant decoders but should be produced as Colour Type 2 by conformant encoders.

B.2 Resolution

The standard resolution is 3142 × 1571 pixels. The hi-res variant is 6284 × 3142 pixels. The dimensions are integer roundings of π×500 and π×1000, so that the nominal angular step is 0.002 rad/pixel at standard resolution and 0.001 rad/pixel at hi-res (see §2.1).

B.3 Depth encoding

The depth value at pixel (u, v) is encoded as:

$$d\_int(u, v) = R(u, v) \times 65536 + G(u, v) \times 256 + B(u, v) \quad (10)$$
$$d(u, v) = d\_int(u, v) \times scale \quad (11)$$

where d(u, v) is the radial distance in metres from the scanner origin to the surface in the direction corresponding to pixel (u, v), and scale is the per-image scaling factor in metres per integer step.

B.4 Required tEXt chunks

The following tEXt ancillary chunks are required:

Key	Value
DepthPano:scale	scaling factor in metres per integer step, as a decimal number
DepthPano:version	encoding version (currently “2.0”).
DepthPano:capture	scanner identifier and capture timestamp (timestamp in ISO 8601)

B.5 Optional tEXt chunks

The following optional values are used when specified by downstream processes.

Key	Default	Value
DepthPano:posePosition	(0,0,0)	Cartesian coordinates of the pose position for the scan (x,y,z).
DepthPano:poseRotation	(0,0,0,1)	rotation of the scan given as a quaternion (w, x, y, z).

Additional tEXt chunks may carry further metadata about the scan capture (scanner model, firmware version, operator identifier). Conformant decoders ignore unknown chunks. Adopters integrating DepthPano into broader bundle architectures (such as the Foundation’s bundle format) will typically add chunks specific to that integration.

B.6 Conformance

A PNG file is a conformant DepthPano file if:

- It is a valid PNG file under ISO/IEC 15948.
- It uses Colour Type 2 (8-bit RGB) without an Alpha channel.
- It uses the tRNS block to specify the transparent value of RGB zero (0,0,0)
- Its resolution is one of the values specified in B.2, or a documented alternative.
- It carries the required tEXt chunks specified in B.4.
- The depth values encoded according to Equations 10–11 produce physically plausible radial distances when interpreted with the scaling factor from the tEXt chunk.

Appendix C. E57 point record structure

E57 is defined by the ASTM E2807 standard. It is a hybrid file format with hierarchical structure and meta data defined in an XML header and point data stored in a binary format. The XML section contains information on the scan and the equipment used, ranges and data types for all fields and which fields are available for each point.

The point cloud information can be represented as a pseudo record constructed from the fields available for each point though the data for each point is not stored as a physical record in the file (fields are grouped together for multiple points to facilitate better packing). Each pseudo point record contains optional components for the following: Cartesian coordinates (X, Y, Z), spherical coordinates (range, elevation and azimuth), intensity, colour (R, G, B), row and column indices, return information and a timestamp. For all components except the indices a state field is provided indicating whether the component is valid for the point. These components map onto the components of the Measurement Stack defined in §2.4.

An example E57 schema (XML) of the pseudo point record is given below. Note, the actual types and ranges of each field are given in the XML header section of the E57 file but provided here for illustrative purpose.

```
<pointRecord type="Structure">
  <!-- Spatial Coordinates (Cartesian or Spherical) -->
  <cartesianX type="Float" precision="single"/>
  <cartesianY type="Float" precision="single"/>
  <cartesianZ type="Float" precision="single"/>

  <sphericalRange type="Float" precision="single"/>
  <sphericalAzimuth type="Float" precision="single"/>
  <sphericalElevation type="Float" precision="single"/>

  <!-- Optical & Reflectivity Attributes -->
  <intensity type="Float" precision="single" minimum="0.0" maximum="1.0"/>
  <colorRed type="Integer" minimum="0" maximum="255"/>
  <colorGreen type="Integer" minimum="0" maximum="255"/>
  <colorBlue type="Integer" minimum="0" maximum="255"/>

  <!-- Scan Structure Metadata -->
  <rowIndex type="Integer" minimum="0" maximum="4000"/>
  <columnIndex type="Integer" minimum="0" maximum="10000"/>

  <!-- Return Information -->
  <returnCount type="Integer" minimum="1" maximum="8"/>
  <returnIndex type="Integer" minimum="0" maximum="7"/>

  <!-- Timing -->
  <timeStamp type="Float" precision="double"/>

  <!-- Validity -->
  <cartesianInvalidState type="Integer" minimum="0" maximum="2"/>
  <sphericalInvalidState type="Integer" minimum="0" maximum="2"/>
  <colorInvalidState type="Integer" minimum="0" maximum="2"/>
  <intensityInvalidState type="Integer" minimum="0" maximum="2"/>
  <timeStampInvalidState type="Integer" minimum="0" maximum="2"/>
</pointRecord>
```

An example JSON payload (an instantiated point) follows. When binary E57 points are parsed and serialised into a JSON message payload, a single point could look like this:


```
{
  "cartesianX": 12.4503,
  "cartesianY": -3.8912,
  "cartesianZ": 1.1047,
  "intensity": 0.724,
  "colorRed": 142,
  "colorGreen": 188,
  "colorBlue": 71,
  "rowIndex": 1204,
  "columnIndex": 4510,
  "returnCount": 1,
  "returnIndex": 0,
  "cartesianInvalidState": 0,
  "timeStamp": 14.802934
}
```

Appendix D. Measurement Stack File Format Comparison

The following tables give a comparison of the file sizes for the other components of the Measurement Stack. Specifically they include the file sizes for the ColourPano and LaserPano for JPEG and PNG.

Table 14: Measurement Stack using JPEG files for ColourPano and LaserPano

	e57 size	e57.xz	DepthPano	ColourPano	LaserPano	Total Stack	File size ratio	Saving
Average	231,990	74,453	7,940	223	435	8,598	11.55	36.27
Minimum	230,865	74,687	9,091	180	494	9,765	13.08	35.64
Maximum	238,954	74,051	7,395	208	408	8,011	10.82	36.56

Table 15: Measurement Stack using PNG files for ColourPano and LaserPano

	e57 size	e57.xz	DepthPano	ColourPano	LaserPano	Total Stack	File size ratio	Saving
Average	231,990	74,453	7,940	3,597	6,738	18,275	24.55	30.94
Minimum	230,865	74,687	9,091	3,246	6,872	19,209	25.72	30.46
Maximum	229,916	75,050	7,676	3,348	6,786	17,811	23.73	31.27

Given that the average Measurement Stacks for PNG images are more than double in size those using JPEG it is recommended that the JPEG images are used for the ColourPano and LaserPano elements of the stack.

References

ASTM International (2021) *ASTM E2807-21: Standard specification for 3D imaging data exchange*. West Conshohocken, PA: ASTM International.

Australian Building Codes Board (2025) *National Construction Code 2025*. Canberra: Australian Building Codes Board.

Deutsch, P. (1996) *DEFLATE compressed data format specification version 1.3*. RFC 1951. Internet Engineering Task Force. Available at: https://doi.org/10.17487/RFC1951.

Dumić, E., Bjelopera, A. and Nüchter, A. (2022) *Dynamic point cloud compression based on projections, surface reconstruction and video compression*. Sensors, 22(1), 197. doi:10.3390/s22010197.

International Organization for Standardization (2004) *ISO/IEC 15948:2004 Information technology — Computer graphics and image processing — Portable Network Graphics (PNG): functional specification*. Geneva: ISO.

International Organization for Standardization (2021) *ISO/IEC 23090-5:2021 Information technology — Coded representation of immersive media — Part 5: Visual volumetric video-based coding (V3C) and video-based point cloud compression (V-PCC)*. Geneva: ISO.

Kohira, K. and Masuda, H. (2017) *Point-cloud compression for vehicle-based mobile mapping systems using portable network graphics*. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, IV-2/W4, pp. 99–106. doi:10.5194/isprs-annals-IV-2-W4-99-2017.

Tong, L. and Bosdriesz, J. (2026) *Measuring what matters: residential accessibility assessment under fiscal pressure*. Metrological Research Foundation.