Designation: E2807 – 11

Standard Specification for 3D Imaging Data Exchange, Version 1.0[1]

This standard is issued under the fixed designation E2807; the number immediately following the designation indicates the year of original adoption or, in the case of revision, the year of last revision. A number in parentheses indicates the year of last reapproval. A superscript epsilon (ε) indicates an editorial change since the last revision or reapproval.

1. Scope

1.1 This specification describes a data file exchange format for three-dimensional (3D) imaging data, known as the ASTM E57 3D file format, Version 1.0. The term "E57 file" will be used as shorthand for "ASTM E57 3D file format" hereafter.

1.2 An E57 file is capable of storing 3D point data, such as that produced by a 3D imaging system, attributes associated with 3D point data, such as color or intensity, and 2D imagery, such as digital photographs obtained by a 3D imaging system. Furthermore, the standard defines an extension mechanism to address future aspects of 3D imaging.

1.3 This specification describes all data that will be stored in the file. The file is a combination of binary and eXtensible Markup Language (XML) formats and is fully documented in this specification.

1.4 All quantities standardized in this specification are expressed in terms of SI units. No other units of measurement are included in this standard.

1.4.1 *Discussion*—Planar angles are specified in radians, which are considered a supplementary SI unit.

1.5 *This standard does not purport to address all of the safety concerns, if any, associated with its use. It is the responsibility of the user of this standard to establish appropriate safety and health practices and determine the applicability of regulatory limitations prior to use.*

1.6 *This standard does not purport to address legal concerns, if any, associated with its use. It is the responsibility of the user of this standard to comply with appropriate regulatory limitations prior to use.*

2. Referenced Documents

2.1 *ASTM Standards:*[2]

E2544 Terminology for Three-Dimensional (3D) Imaging Systems

2.2 *IEEE Standard:*[3]

754-1985 IEEE Standard for Binary Floating-Point Arithmetic

2.3 *IETF Standard:*[4]

RFC 3720 Internet Small Computer Systems Interface (iSCSI)

2.4 *W3C Standard:*[5]

XML Schema Part 2: Datatypes Second Edition

3. Terminology

3.1 *Definitions*—Terminology used in this specification conforms to the definitions included in Terminology E2544.

3.2 *Definitions of Terms Specific to This Standard:*

3.2.1 *backward compatibility, n*—ability of a file reader to understand a file created by a writer of an older version of a file format standard.

3.2.2 *byte, n*—grouping of 8 bits, also known as an octet.

3.2.3 *camel case, n*—naming convention in which compound words are joined without spaces with each word's initial letter capitalized within the component and the first letter is either upper or lowercase.

3.2.4 *camera image, n*—regular, rectangular grid of values that stores data from a 2D imaging system, such as a camera.

3.2.5 *camera projection model, n*—mathematical formula used to convert between 3D coordinates and pixels in a camera image.

3.2.6 *file offset, n*—see *physical file offset*.

3.2.7 *file-level coordinate system, n*—coordinate system common to all 2D and 3D data sets in a given E57 file.

3.2.8 *forward compatibility, n*—ability of a file reader to read a file that conforms to a newer version of a format specification than it was designed to read, specifically having the capability to understand those aspects of the file that were defined in the version it was designed to read, while ignoring those portions that were defined in later versions of the format specification.

---

[1] This specification is under the jurisdiction of ASTM Committee E57 on 3D Imaging Systems and is the direct responsibility of Subcommittee E57.04 on Data Interoperability.

Current edition approved Feb. 1, 2011. Published March 2011.

[2] For referenced ASTM standards, visit the ASTM website, www.astm.org, or contact ASTM Customer Service at service@astm.org. For *Annual Book of ASTM Standards* volume information, refer to the standard's Document Summary page on the ASTM website.

[3] For referenced IEEE standards, visit http://grouper.ieee.org/groups/754.

[4] For referenced Internet Engineering Task Force (IETF) standards, visit the IETF website, www.ietf.org.

[5] String representations (the lexical space) of the numeric datatypes are documented in the W3C standard: "XML Schema Part 2: Datatypes Second Edition", available on the website http://www.w3.org/TR/xmlschema-2/.

3.2.9 *logical length, n*—number of bytes used to describe some entity in an E57 file, not including CRC checksum bytes.

3.2.10 *physical file offset, n*—number of bytes preceding the specified byte location in an E57 file, counting payload bytes and checksums.

3.2.10.1 *Discussion*—This term is also known as the *file offset*.

3.2.11 *physical length, n*—number of bytes used to describe some entity in an E57 file, including CRC checksum bytes.

3.2.12 *record, n*—single collection in a sequence of identically-typed collections of elements.

3.2.13 *rigid body transform, n*—type of coordinate transform that preserves distances between all pairs of points that furthermore does not admit a reflection.

3.2.13.1 *Discussion*—A rigid body transform can be used, for example, to convert points from the local coordinates of a 3D data set (for example, a single laser scan) to a common coordinate system shared by multiple 3D data sets (for example, a set of laser scans).

3.2.14 *XML namespace, n*—method for qualifying element names in XML to prevent the ambiguity of multiple elements with the same name.

3.2.14.1 *Discussion*—XML namespaces are used in an E57 file to support the definition of extensions.

3.2.15 *XML whitespace, n*—sequence of one or more of the following Unicode characters: the space character (20 hexadecimal), the carriage return (0D hexadecimal), line feed (0A hexadecimal), or tab (09 hexadecimal).

3.2.16 *zero padding, n*—one or more zero-valued bytes appended to the end of a sequence of bytes.

4. Acronyms

4.1 *ASCII*—American Standard Code for Information Interchange

4.2 *CRC*—Cyclic redundancy check

4.3 *GUID*—Globally unique identifier

4.4 *IEEE*—Institute of Electrical and Electronics Engineers

4.5 *IETF*—Internet Engineering Task Force

4.6 *iSCSI*—Internet small computer system interface

4.7 *JPEG*—Joint Photographic Experts Group

4.8 *PNG*—Portable network graphics

4.9 *URI*—Uniform resource identifier

4.10 *UTC*—Coordinated universal time

4.11 *UTF*—Unicode Transformation Format

4.12 *W3C*—WorldWide Web Consortium

4.13 *XML*—eXtensible Markup Language

5. Notation and Mathematical Concepts

5.1 The following notation and established mathematical concepts are used in this specification.

5.2 *Intervals:*

5.2.1 A closed interval is denoted $[a, b]$, where $a \le b$. A closed interval includes the endpoints a and b and all numbers in between. An open interval is denoted (a, b), where $a \le b$. An open interval includes the numbers between the endpoints a and b, but does not include the endpoints themselves. The half-open intervals $(a, b]$ and $[a, b)$ do not include the a and b endpoints, respectively.

5.3 *Cartesian Coordinate System:*

5.3.1 Points in Cartesian coordinates are represented by an ordered triplet (x, y, z), where x, y and z are coordinates along the X, Y, and Z axes, respectively. The coordinate system is right-handed.

5.4 *Cylindrical Coordinate System:*

5.4.1 Points in cylindrical coordinates are represented by an ordered triplet (ρ, θ, z), where ρ is the radial distance (in meters), θ is the azimuth angle (in radians), and z is the height (in meters).

5.4.1.1 The azimuth angle is measured as the counterclockwise rotation of the positive X-axis about the positive Z-axis of a Cartesian reference frame.

5.4.2 The following restrictions on cylindrical coordinates are applied:

$$\rho \ge 0 \tag{1}$$

$$-\pi < \theta \le \pi \tag{2}$$

5.4.3 The conversion from Cartesian to cylindrical coordinates is accomplished through the formulas (note that the z coordinate is the same in both systems):

$$\rho = \sqrt{(x^2 + y^2)} \tag{3}$$

$$\theta = \text{atan2}(y, x) \tag{4}$$

5.4.3.1 The function "atan2(y, x)" is defined as the function returning the arc tangent of y/x, in the range $(-\pi, +\pi]$ radians. The signs of the arguments are used to determine the quadrant of the result.

5.4.3.2 In degenerate cases, the following convention is observed:

If $x = y = 0$, then $\theta = 0$.

5.4.4 Conversely, cylindrical coordinates can be converted to Cartesian coordinates using the formulas:

$$x = \rho \cos(\theta) \tag{5}$$

$$y = \rho \sin(\theta) \tag{6}$$

5.5 *Spherical Coordinate System:*

5.5.1 Points in spherical coordinates are represented by an ordered triplet (r, θ, φ), where r is the range (in meters), θ is the azimuth angle (in radians), and φ is the elevation angle (in radians).

5.5.2 The following restrictions on spherical coordinates are applied:

$$r \ge 0 \tag{7}$$

$$-\pi < \theta \le \pi \tag{8}$$

$$-\frac{\pi}{2} \le \varphi \le \frac{\pi}{2} \tag{9}$$

5.5.3 The conversion from spherical to Cartesian coordinates is accomplished through the formulas:

$$x = r\cos(\varphi)\cos(\theta) \quad (10)$$

$$y = r\cos(\varphi)\sin(\theta) \quad (11)$$

$$z = r\sin(\varphi) \quad (12)$$

5.5.4 Conversely, in non-degenerate cases, Cartesian coordinates can be converted to spherical coordinates via the formulas:

$$r = \sqrt{(x^2+y^2+z^2)} \quad (13)$$

$$\theta = \mathrm{atan2}(y,x) \quad (14)$$

$$\varphi = \arcsin\left(\frac{z}{r}\right) \quad (15)$$

5.5.4.1 In degenerate cases, the following conventions are observed:

If $x = y = 0$, then $\theta = 0$;

If $x = y = z = 0$, then both $\theta = 0$ and $\varphi = 0$.

5.5.5 *Discussion*—The elevation is measured with respect to the *XY*-plane, with positive elevations towards the positive *Z*-axis. The azimuth is measured as the counterclockwise rotation of the positive *X*-axis about the positive *Z*-axis. This definition of azimuth follows typical engineering usage. Note that this differs from traditional use in navigation or surveying.

5.6 *Quaternions:*

5.6.1 A quaternion is a generalized complex number. A quaternion, q, is represented by an ordered four-tuple (w,x,y,z), where $q = w + x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$. The coordinate w defines the scalar part of the quaternion, and the coordinates (x, y, z) define the vector part.

5.6.2 The norm of a quaternion, $\| q \|$, is defined as:

$$\| q \| = \sqrt{w^2+x^2+y^2+z^2}.$$

5.6.3 A unit quaternion, q, has the further restriction that its norm $\| q \| = 1$.

5.6.4 Rotation of a point $\boldsymbol{p}$ by a unit quaternion q is given by the matrix formula:

$$\boldsymbol{p}' = \boldsymbol{Rp} \quad (16)$$

where:

$$\boldsymbol{R} = \begin{bmatrix} w^2+x^2-y^2-z^2 & 2(xy-wz) & 2(xz+wy) \\ 2(xy+wz) & w^2+y^2-x^2-z^2 & 2(yz-wx) \\ 2(xz-wy) & 2(yz+wx) & w^2+z^2-x^2-y^2 \end{bmatrix} \quad (17)$$

5.6.5 *Discussion*—Unit quaternions are used in this standard to represent rotations in rigid body transforms.

5.7 *Rigid Body Transforms:*

5.7.1 A rigid body transform converts points from one coordinate reference frame to another, preserving distances between pairs of points and, furthermore, not admitting a reflection. A rigid body transform can be represented as a 3×3 rotation matrix $\boldsymbol{R}$ and a translation 3-vector $\boldsymbol{t}$.

5.7.2 A 3D point is transformed from the source coordinate system to the destination coordinate system by first applying the rotation and then the translation. More formally, the transformation operation *T(.)* of a point $\boldsymbol{p}$ is defined as:

$$\boldsymbol{p}' = T(\boldsymbol{p}) = \boldsymbol{Rp}+\boldsymbol{t} \quad (18)$$

The rotation matrix $\boldsymbol{R}$ can be computed from a unit quaternion q using Eq 17.

5.7.3 *Discussion*—Rigid body transforms are used in this standard to support the transformation of data represented in a local coordinate system, such as the coordinate system of a sensor used to acquire a 3D data set, to a common file-level coordinate system shared by all 3D data sets.

5.8 *Trees:*

5.8.1 A tree is data structure that represents an acyclic graph. A tree consists of nodes, which store some information, and edges (also known as arcs) that connect the nodes. The single topmost node is called the root node. A node may have zero or more nodes connected below it, which are called child nodes. Each node, except the root node, has exactly one node connected above it, which is called the parent node. Nodes with no children are called leaf nodes. A descendant is a direct or indirect child of a given node.

5.8.2 *Discussion*—Trees are used in this standard to describe the structure of XML data, as well as index data in binary sections.

5.9 *XML Elements and Attributes :*

5.9.1 An XML element is the fundamental building block of an XML file. An element consists of a start tag, optional attributes, optional child elements, optional child text, and an end tag. Element names in an E57 file are case sensitive. Element names in this specification are written in camel case with a lowercase initial character. Type names in this specification are written in camel case with an upper case initial character.

5.9.2 *Discussion*—See Fig. 1 for an excerpt of XML that illustrates the parts of an XML element.

5.9.3 XML elements that have child elements form a tree, with each element being a node.

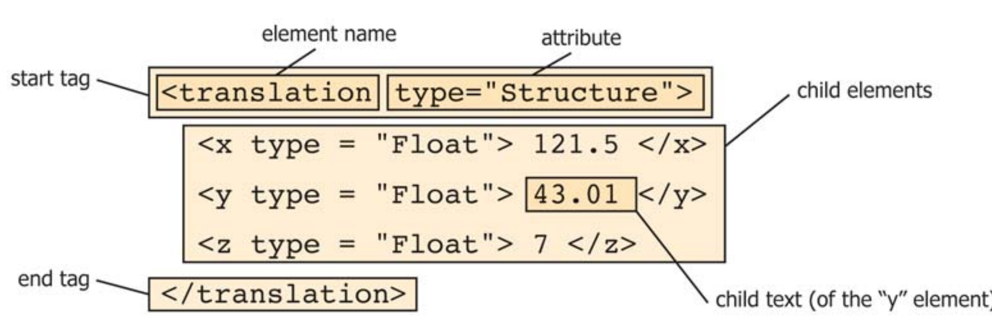

FIG. 1 XML Elements and Attributes