# The C3D rotation data format

This is an unofficial update of the description of the ROTATION data type format proposed by C-Motion in 2019 to record a maximum of 255 location data samples, each stored as a 4x4 transformation matrix to store 6 DOF sample data from each sensor or subject segment, generated in a motion capture environment, together with an additional number indicating the reliability of each measurement. Each location data sample consists of a 4x4 rotation matrix that transforms a signal from the local coordinates of the sensor/segment to the data collection volume coordinates. Since the proposed ROTATION storage format is not the standard 3D Point data format (defined when the second byte of the C3D file header is 0x50h, an ASCII “P” char), all standard C3D applications fail to read the ROTATION data because the C3D file header defines the format used to store data in the C3D file parameter/data sections.

The stored data samples in the ROTATION format can be used to define landmarks and segments, and track segments not individual points so the ROTATION format is unique and needs to be stored in a unique C3D file data section defined by setting the second byte of the C3D header block to 0x52h (ASCII “R” character), for example:

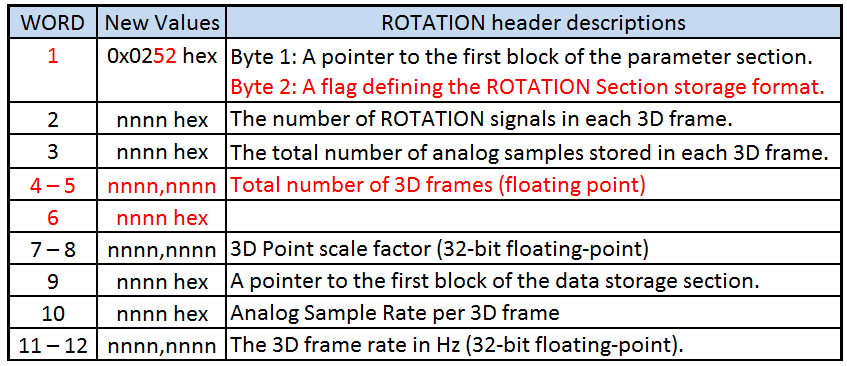


Figure 7- A potential C3D Rotation format file header with updates in red.

All applications reading C3D files will need relatively minor updates to read the new C3D Rotation format. The concept of the ROTATION header proposal is that it enables applications to determine if the C3D file data section contains 3D Point data or Rotation data and describes the data section contents in a standard C3D structure.

This change to the C3D file header will prevent existing applications that are only designed to read 3D Point data from opening C3D files containing the ROTATION format, however while applications can open existing C3D file that store the ROTATION formatted data section that is flagged as 3D Point data by header byte #2, they cannot read the data. The ROTATION format is a new C3D data storage method, applications need to be able to instantly identify the format of a file when they open it.

Attempting to define a deviation from the traditional 3D Point format definition by creating a few parameters and modifying the internal storage allocations will cause problems when applications do not realize that the format has changed but attempt to update the file to fix the format error that they perceive, or simply add a few new parameters.

An additional failure in the C-Motion format is that their C3D file only supports floating-point format, it does not support integer data storage and the support for the C3D endian format is undocumented.

## The ROTATION Group

The ROTATION parameters group provides information about the rotation data stored in the associated data section. As a result, the parameters ROTATION:USED, ROTATION:DATA\_START, ROTATION:RATE, ROTATION:FRAMES, and ROTATION:LABELS are required when the C3D file data section stores 6 DOF data. Other ROTATION parameters may be required by specific software applications.

Note that all C3D parameter and group structures have an associated description string that should be used to provide basic information about each group and parameter.

### ROTATION:USED

The ROTATION:USED parameter is an unsigned integer that contains the number of ROTATION signals (1 to 65535) that are written to each frame of data in the C3D file data section. This parameter enables the ROTATION data section of the file to be interpreted and its size calculated. Every ROTATION signal stored in a C3D file is described by an associated entry in the ROTATION:LABELS and documented by the related ROTATION:DESCRIPTIONS parameters.

### ROTATION:DATA\_START

The ROTATION:DATA\_START parameter is an unsigned 16-bit integer value used as a pointer to the first block of the data section within the C3D file parameter/data section and must always be used to determine location of the data section. A C3D file block is always 512 bytes long (256 sixteen-bit words).

A copy of the ROTATION:DATA\_START parameter value can also be found stored in word 9 of the associated C3D file header record to enable software applications to quickly locate the start of 3D data. The DATA\_START value stored in the header record must always be identical to the parameter value. As a result of this parameter being stored as an integer in the C3D file header, it must always be written as an integer value in the parameter section.

### ROTATION:RATE

The ROTATION:RATE parameter is a floating-point value storing the ROTATION sample rate of the data contained within the C3D file in samples per second. If the ROTATION data is sampled at a rate of 60 samples per second then RATE should be set to 60. It is important that the ROTATION:RATE parameter is accurately recorded as it is used to calculate timing for the ROTATION data samples and affects the sample rate of analog data stored in the ROTATION data section which is always an integer multiple of the ROTATION:RATE parameter.

A copy of the ROTATION:RATE parameter value can also be found stored in floating-point format in words 11-12 of the C3D file header, this value must be an identical copy of the value stored in the parameter section.

### ROTATION:FRAMES

The ROTATION:FRAMES parameter is single value that stores the number of ROTATION data frames that are recorded in the C3D file and can be stored as either an integer or a floating-point value. The unsigned integer format is recommended when the frame count is less 65535, although the FRAME count is stored in the related header record as a floating-point value in words 4-5. Note that the first frame in the frame count is frame number one (there is no “frame zero”) and if the ROTATION frame count needs to be synchronized with external data files then the TRIAL group can store the external frame range in the ACTUAL\_START\_FIELD and ACTUAL\_END\_FIELD parameters for external synchronization.

### ROTATION:LABELS

The ROTATION:LABELS parameter is a character data array that consisted of a unique ASCII value for each ROTATION sample stored in the C3D file data section. The LABELS array values are usually four characters of upper-case ASCII text (A-Z, underscore, and 0-9) although longer labels and UTF-8 encoding are permitted. Each label (HIPS, SAGITAL, FRONTAL etc.) is used to provide a unique reference within the C3D file data section. This allows software applications to identify and process data based on the unique label identification because the order of the samples stored in the data section may vary from one data capture to another.

The purpose of the LABELS parameter is to allow applications reading data from the C3D file to search for a specific ROTATION sample by referencing its LABELS value. This allows applications to be written from multiple files referencing the ASCII identification because the order to the data sample may vary from one file to another. An application that references the LABELS will work in any environment, as it does not require that the data is stored in any specific order within the C3D file.

The ROTATION:LABELS strings should not normally exceed 16 characters; their function is to provide a unique, machine readable reference, not a detailed human readable description.

Individual labels must always be unique to identify each sample in the file but there is no need to make them descriptive as the ROTATION:DESCRIPTIONS parameter is provided for human intelligible descriptions. It is recommended that LABELS are always no more than 16 characters in length. This parameter is not normally locked and may be edited if necessary – editing the labels only changes the ASCII reference that identifies a specific movement and does not affect the C3D file structure.

### ROTATION:DESCRIPTIONS

The ROTATION:DESCRIPTIONS parameter is a character data array that provides a human readable description of each recorded ROTATION sample, internally referenced by the associated ROTATION:LABELS parameter. There should always be a one to one relationship between the number of LABELS and the number of DESCRIPTIONS parameters.

Each DESCRIPTIONS parameter can be 255 8-bit characters long and can use UTF-8 encoding to support localized character sets. Each stored descriptions should be unique and document the associated LABELS sample. This parameter exists to provide human readable documentation about each of the individual ROTATION samples referenced by ROTATION:LABELS, which are concise machine readable data sample identifications, e.g. LLEG. The associated description might be “Left leg IMU sensor #5” and should normally be unique. The parameter is not locked and may be edited without affecting the C3D file structure.

## The C3D ROTATION data format

Each rotation signal is stored as 17 floating-point values. The first 16 floating-point values describe a 4x4 ROTATION matrix (stored as columns) followed by a single floating-point value that records the “reliability metric” value.

As a result of the ROTATION matrix being stored as floating-point values, all related analog data samples in the data section must also be stored as floating-point values in synchronization with each ROTATION frame of data. The analog sample rate must be an integer multiple (x1, x2, x5 etc.) of the ROTATION frame rate to guarantee data synchronization using the same method and ANALOG parameters defined by the C3D point format (ID 0x50h).

The C3D Rotation format stores a maximum of 255 location data samples, each written as a 4x4 transformation matrix, storing six DOF sample data from each sensor or subject segment generated in the motion capture environment. Each location data sample consists of a 4x4 rotation matrix that transforms a signal from the local coordinates of the sensor/segment to the laboratory data collection volume coordinates stored in millimeters, the C3D measurement standard.

[R4x4] [Vlocal4x1] = [Vlab4x1]

The signals are stored in column major format in the C3D file data section but the function of the individual matrix values has not been documented, a “reliability metric” is stored in associated with each 4x4 matrix but its function in relation to each 4x4 matrix sample is not documented.

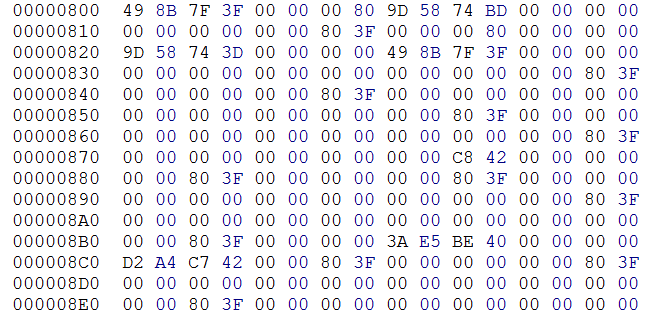


Figure 8 – The raw C-Motion ROTATION data section format containing no analog data.

While restricting the data section to floating-point formatted data does not break the C3D file format it adds a complication to the interpretation of the C3D format and results in files that are twice the size on a scaled integer format, requiring additional processing when a file is read. The interpretation of the first two rotation samples from the data section shown above illustrates these issues assuming that the undocumented structure stores the undocumented reliability metric value as the 17th floating-point sample in the structure.

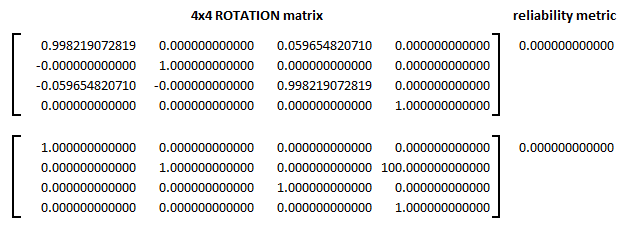


Figure 9 – Two rotation data samples stored in the first 136 bytes of the data section.

Essentially the problem with the C3D Rotation format described by C-Motion is that it has only been described but not fully documented so users cannot access the data with MATLAB or other data access and programming environments. Simply reading the stored values does not explain their function.