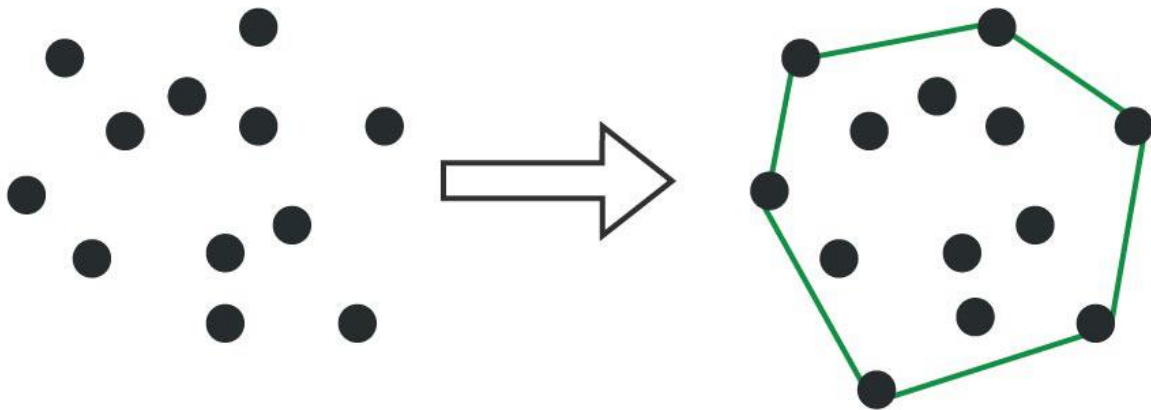


Metric Compute Volume of a Convex Hull Across Frames

Overview

The metric command **Compute Volume of a Convex Hull Across Frames** can be used to find the area or volume of the convex hull for a dataset across several frames. A convex hull is defined as the smallest convex set containing all of the points in a Euclidian space. It can be visualized in 2 dimensions as a string or rubber band around the perimeter of a set, or in 3 dimensions as shrink wrap enclosing the entire range of data points.



Convex Hull



The volume or area of the convex hull of a set can be used to describe the amount of variance in a set or to describe the geometry of the distribution of points.

Pipeline Command

The pipeline command for **Compute Volume of a Convex Hull Across Frames** can be found in the **Metric** folder in the Pipeline Workshop and is as so:

```
Metric_Compute_Volume_of_a_Convex_Hull_Across_Frames
/SIGNAL_TYPES=
! /SIGNAL_FOLDER=ORIGINAL
! /SIGNAL_NAMES=
! /RESULT_FOLDER=PROCESSED
/RESULT_NAME=
! /NUMBER_OF_DIMENSIONS=
! /COMPONENT_SEQUENCE=
! /REQUIRE_ALL_SIGNALS=TRUE
! /START_FRAME=
! /END_FRAME=
```

```
! /EVENT_SEQUENCE=  
! /EXCLUDE_EVENTS=  
;
```

Command Parameters

The following table shows the command parameters seen above and their descriptions:

SIGNAL_TYPES	Specify the signal type
SIGNAL_FOLDER	Specify the origin folder
SIGNAL_NAMES	Specify the Signal to be used
RESULT_FOLDER	The name of the result signal folder
RESULT_NAME	The name of the result signal
NUMBER_OF_DIMENSIONS	Number of dimensions the hull will be computed for (2 or 3)
COMPONENT_SEQUENCE	Specify the Signal components to be used (e.g. X + Y + Z or 0 + 1 + 2 etc)
REQUIRE_ALL_SIGNALS	True or False: require all signals to exist
/START_FRAME=	The frame at which to start recording data.
END_FRAME=	The frame at which to stop recording data.
EVENT_SEQUENCE	A list of events (separated by “+” signs). For example, LHS+RTO
EXCLUDE_EVENTS	If this event occurs before the first and last event, do not computed a metric

Dialog

The command can be edited in a text editor or in a dialog form. To edit in the dialog pop up form either click on the **Edit** button in the pipeline workshop or double-click on the pipeline command. The dialog is shown below:

Compute the Convex Hull of a set of Vertices using Data from a Range of Frames

The Resulting Metric Signal

Result Name

Result Folder

Specify the Signal Components

Signal Component (X,Y,Z,R) or (1,2,...) or ALL

Signal Requirements

Require All Signals to Exist

Select Range by Expression

Start Frame

End Frame

Select Range by Events

Defined Events		Event Sequence
Event	>>	
LHS	<<	
LOFF		
LON		
LTO		
RHS		
ROFF		
RON	>>	
RTO	<<	

Exclude sequences containing these events

The dialog box allows you to assign values to the command parameters outlined above

Example: 2D convex hull

Here the **Metric Compute Volume of a Convex Hull Across Frames** command is used to compute the convex hull for right and left foot GRF data during a balance trial. By computing a 2 dimensional (X and Y) hull for the GRF at each foot we can make comparisons between sides while balancing and identify potential asymmetry.

The command is as so:

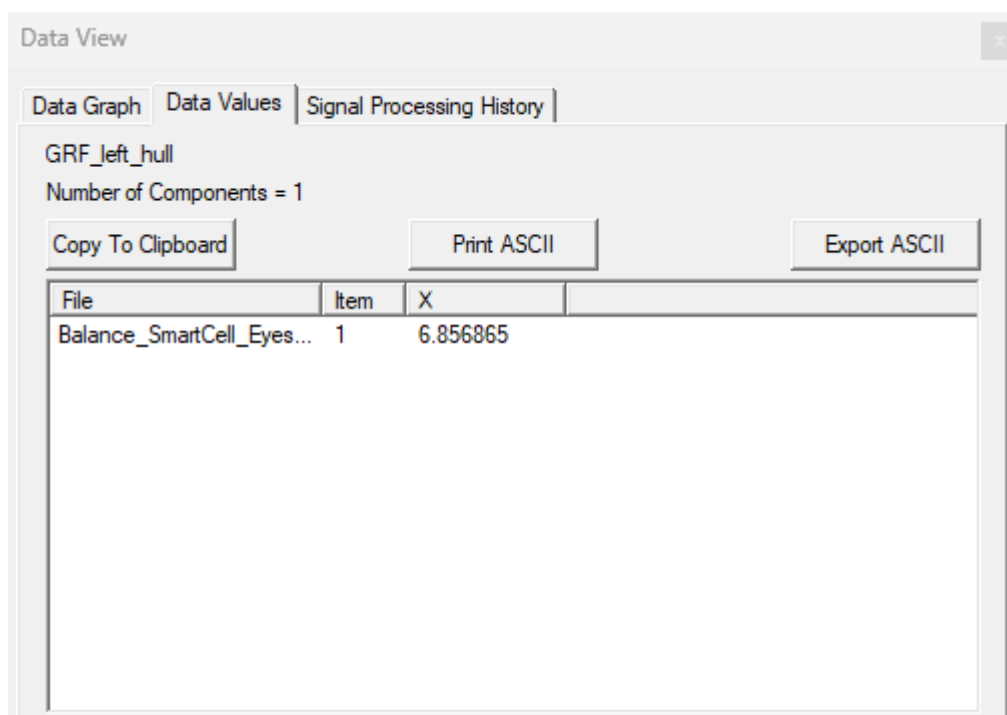
```
Metric_Compute_Volume_of_a_Convex_Hull_Across_Frames
/SIGNAL_TYPES=FORCE
/SIGNAL_FOLDER=ORIGINAL
/SIGNAL_NAMES=FP3
/RESULT_FOLDER=PROCESSED
/RESULT_NAME=GRF_right_hull
/NUMBER_OF_DIMENSIONS=2
/COMPONENT_SEQUENCE=X+Y
/REQUIRE_ALL_SIGNALS=TRUE
```

```
! /START_FRAME=
! /END_FRAME=
/EVENT_SEQUENCE=Start+End
! /EXCLUDE_EVENTS=
;

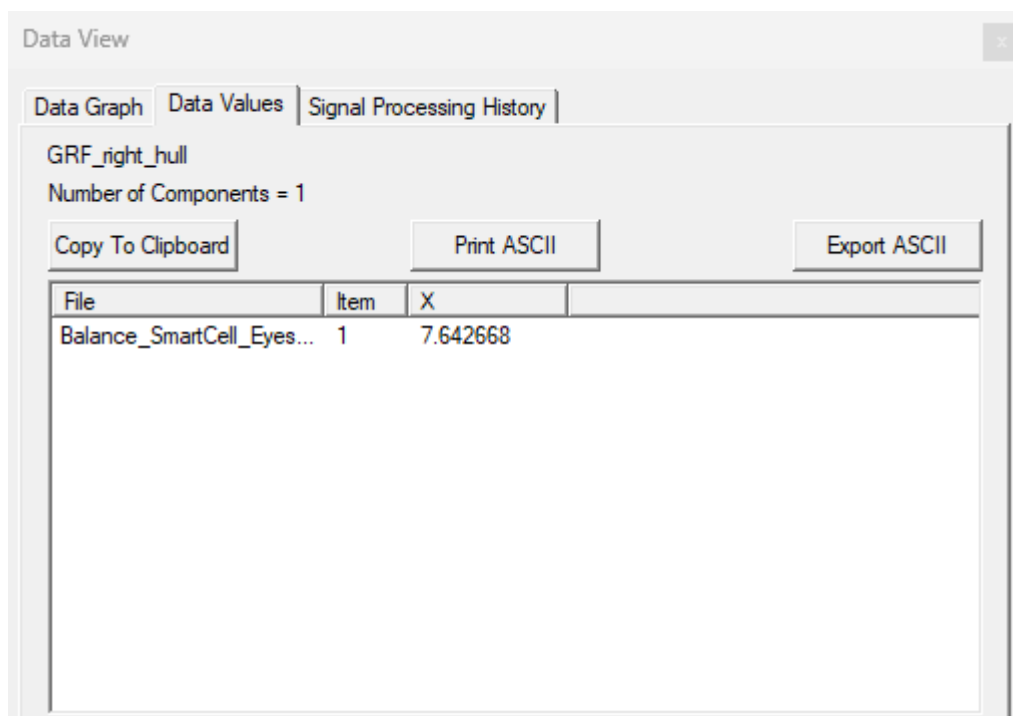
Metric_Compute_Volume_of_a_Convex_Hull_Across_Frames
/SIGNAL_TYPES=FORCE
/SIGNAL_FOLDER=ORIGINAL
/SIGNAL_NAMES=FP4
/RESULT_FOLDER=PROCESSED
/RESULT_NAME=GRF_left_hull
/NUMBER_OF_DIMENSIONS=2
/COMPONENT_SEQUENCE=X+Y
/REQUIRE_ALL_SIGNALS=TRUE
! /START_FRAME=
! /END_FRAME=
/EVENT_SEQUENCE=Start+End
! /EXCLUDE_EVENTS=
;
```

The resulting metrics show the area of the 2 dimensional hull created for each side, a larger area points to more variation in force during the trial.

Results:



File	Item	X
Balance_SmartCell_Eyes...	1	6.856865



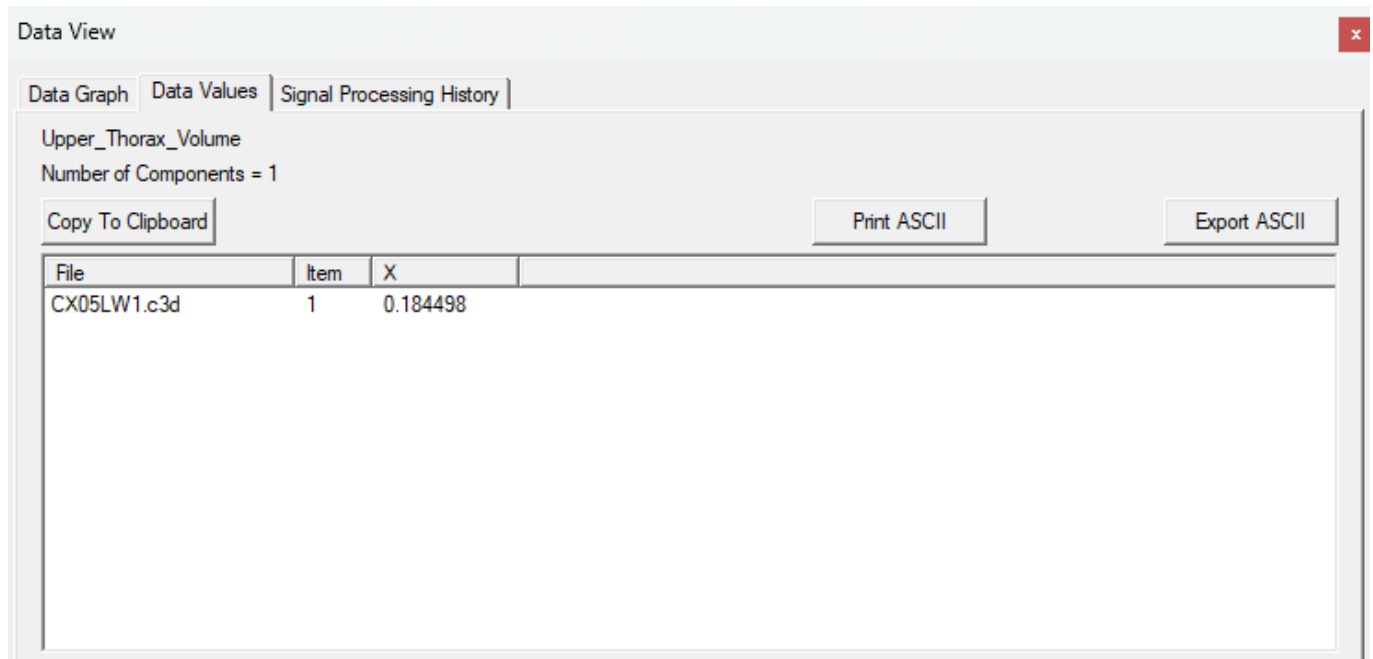
Example: 3D convex hull

In this example **Metric Compute Volume of a Convex Hull Across Frames** is used to find the volume occupied by an individual's thorax segment between events as defined in the workspace.

The pipeline command is as follows:

```
Metric_Compute_Volume_of_a_Convex_Hull_Across_Frames
/SIGNAL_TYPES=TARGET
! /SIGNAL_FOLDER=ORIGINAL
/SIGNAL_NAMES=CLAV+STRN+CV7+T10+RSH0+LSH0
! /RESULT_FOLDER=PROCESSED
/RESULT_NAME=Upper_Thorax_Volume
/NUMBER_OF_DIMENSIONS=3
/COMPONENT_SEQUENCE=ALL
! /REQUIRE_ALL_SIGNALS=TRUE
! /START_FRAME=
! /END_FRAME=
/EVENT_SEQUENCE=EVENT1+EVENT2
! /EXCLUDE_EVENTS=
;
```

Here a 3 dimensional hull is used as the resulting metric will represent a 3 dimensional structure in space. The resulting metric value represents the approximate volume of the space that the upper thorax segment travels through between the events listed:



Additional Resources:

[Delaunay Triangulation](#)
[Convex Hull Algorithm](#)

From:
<https://www.wiki.has-motion.com/> - **Wiki Documentation Site**

Permanent link:
https://www.wiki.has-motion.com/doku.php?id=metric_convex_hull_across_frames

Last update: **2025/10/06 19:52**

