

Packet Formats

Michael Norris

May 6, 2016

What's In This Document

This document describes the packet formats for the software modules that we have developed. These packets are used for communicating data to the Python visualization module from Platform.cpp

SeptoServer Packet

SeptoServer (code that was provided to us by Narges) transmits packets of data to the code that we have written. Originally, SeptoServer wrote these packets into a file. We added an http server and packet queue in SeptoServer that serves the most recent packets to The data values in the packets are defined below:

```
toolname: string
framenum: string
timestamp: string of a long, of a unix timestamp, seconds since Unix epoch
portHandleStatus: string of the status, \\ portHandleStatus be OK if the tool is
x: string of the x component translation of the transformation
y: string of the y component translation of the transformation
z: string of the z component translation of the transformation
r00: string of double, element 00 of the rotation matrix
r01: string of double, element 01 of the rotation matrix
r02: string of double, element 02 of the rotation matrix
r10: string of double, element 10 of the rotation matrix
r11: string of double, element 11 of the rotation matrix
r12: string of double, element 12 of the rotation matrix
r20: string of double, element 20 of the rotation matrix
r21: string of double, element 21 of the rotation matrix
r22: string of double, element 22 of the rotation matrix
stat: Root Mean Square error of the reading
```

Table 1. SeptoServer packet data description.

Each packet is a single ascii-encoded string. Each value in the packet is separated by a tab character. If there is one sensor, then these 17 values will appear in the order listed above, separated by tabs, and will terminate with a newline character. If there are n sensors, then these 17 values will appear in the order listed above, with $n*17$ values in the packet.

If there are 3 sensors (which our project requires), then each packet will contain 51 data elements, each separated by tabs, and will terminate with a newline character. Entries 1 – 17 come from sensor 1, 18 – 34 come from packet 2, etc.

SeptoServer Packet Reception

The code written in Platform.cpp processes the packets from the SeptoServer. We require that sensor 1 is the sensor placed on the patient’s forehead, sensor 2 is the sensor attached to the surgical scissors, and sensor 3 is the pointer.

Platform.cpp processes the packets in three modes.

0.1 Input Modes

0.1.1 pipe

When the pipe flag is passed to the executable created from Platform.cpp, packets are read from standard input.

0.1.2 http

When the http flag is passed to the executable, packets are read from a tcp client.

0.1.3 file

When neither of these flags are specified, the command line argument is assumed to be the name of a file of a recorded session which is read line-by-line.

0.2 Processing Modes

There are two algorithms which are run on the EM sensor pose data. The Processing Mode determines where the data packets from the SeptoServer are sent by the Platform object.

A GUI will allow the user to choose the current mode. One button will say "Septum Tracing Mode," and the other button will say "Line of Cut Mode." Only one mode can be selected at a time. The selected mode should have visual indication that it has been selected.

Septum Tracing Mode

This mode will generate the Septum Surface Model, which is used in the Line of Cut Mode. The Line of Cut Mode cannot run until the Septum Surface Model has been created.

During the Septum Tracing Mode, the packet data is turned into transformComponent objects. The packet from sensor 1 (the patient's forehead sensor) and the packet from sensor 3 (the pointer) are placed into containers. The packet from sensor 2 is dropped. When Septum Tracing Mode ends, a Surface model will be generated and sent to the visualization module using the Septum Surface Generation algorithm defined in the Septum Tracing Procedure.

Septum Tracing Mode produces packets described in the next section.

Line of Cut Mode

When the Septum Tracing Mode has generated a Septum Surface Model and transmitted it to the visualization module, the Line of Cut Mode can begin.

Line of Cut Mode will take data from sensor 1 (the patient's forehead sensor) and sensor 2 (the scissor sensor) and perform the Line of Cut Prediction algorithm. The data from sensor 1 and sensor 2 in each packet will be fed to the line of cut prediction algorithm and will be written to standard output. The visualization module's standard input is piped from the output of the Platform module, so the visualization module is able to receive the Line of Cut Packets.

Packets for Visualization

When the Septum Tracing Mode has completed, a Septum Surface packet needs to be sent to the visualization module. These packets are transmitted by printing to standard output. The visualization module is connected via a unix pipe to the standard output of the Platform class that controls the line of cut creation and septum surface creation.

When the Line of Cut Mode is active, a Line of Cut packet is sent to the visualization module.

We define a standard packet form in the first section, then outline the Septum Surface packets and Line of Cut packet formats.

Standard Packet Form

Each data packet will be an ascii-encoded string with fields separated by tabs, and each packet will terminate in newline characters. Multiple packets can be joined together through string concatenation provided each packet ends in a newline character.

Each packet begins with an integer specifying the packet type. The packet types are:

Integer Prefix	Packet Type
1	Septum Plane
2	Septum Surface Reset Packet
3	Septum Surface Points
4	Septum Surface Vertex Points
5	Line of Cut
6	Septum Convex Hull
7	Training Line Packet
8	Remove Training Line
9	Plane of Cut Packet
A	Training Plane Packet

Septum Plane Packet

The first iteration of our project assumes the Nasal Septum is a mathematical plane. The Septum Plane packet will contain three points, p1, p2, p3. The format of this packet is:

packet prefix	x1	y1	z1	x2	y2	z2	x3	y3	z3
1	x1	y1	z1	x2	y2	z2	x3	y3	z3

Septum Plane Packet

Septum Surface Reset Packet

When the Septum Tracing Mode ends, the Platform will send the visualization module a Septum Surface Reset packet. This packet signifies that the old Septum Surface Model should be deleted.

This packet is the literal string:

```
2\tERASE\n
```

Literal String of Septum Surface Reset Packet

Note: This packet is not implemented. Any old septum surface data is removed when new septum surface data is received.

packet prefix	message
2	ERASE

Tabular form of Septum Surface Reset Packet

The only valid message is ERASE (which deletes the old septum surface model). Any other message is currently undefined.

In the future, this system could be extended to support a mode which stores old models in the visualization module and allow the user to toggle between surface models.

Septum Surface Point Packet

The Septum Surface Point packet will contain points on the surface of the septum mesh. This is a packet that begins with the integer 3, and contains n points collected by the EM pointer.

The packet will contain the integer 3, and interleaved points (x1, y1, z1, x2, y2, z2, ..., xn, yn, zn), which are string representations of double values. All values are separated by tabs, and the whole packet ends in a newline character.

```
3\tx1\ty1\tz1\tx2\ty2\tz2\t...xn\tyn\tzn\n
```

Septum Surface Vertex Points Packet

A Vertex Point Packet contains triplets of the points that constitute a single triangle in the Septum Surface mesh. The format of the packet contains the packet prefix, followed

by tab-spaced triplets of string representations of integers.

All values in the packet are separated by tabs, and terminate in a newline character.

packet prefix	int	int	int	...	int	int	int	newline
4	p1	p2	p3	...	p_{n-2}	p_{n-1}	p_n	newline

Line of Cut Packet

The Line of Cut Packet contains the packet prefix 5, the scissor pose, the start of the line of cut, the end of the line of cut, a timestamp, and a flag for whether the line of cut is valid (whether it intersects the septum plane). All values are tab spaced, and the packet ends in a newline.

The scissor rotation is given in three Euler angles, it's the scissor rotation relative to the septum plane, the scissor translation is three doubles, the line of cut start is a vector with three doubles, the line of cut end is a vector with three doubles, the timestamp is a long since unix epoch, and valid is a flag that is 0 if the line of cut is not in the septum plane, and 1 otherwise.

The packet starts with the packet prefix, and contains all fields in the order given, they are tab-spaced, and it ends in a newline.

Septum Convex Hull Packet

The Septum Convex Hull packet begins with a 6, and contains a list of 3 or more (x,y,z) points on the convex hull. All values are tab spaced and the packet ends in a newline character.

Training Line Packet

The Training Line Packet begins with a 7, and contains two (x,y,z) triplet. The first (x,y,z) triplet is the start of the line, and the second one is the end of the line. The packet's format is 7, tab, x, tab, y, tab, z, tab, x, tab, y, tab, z, end

Remove Training Line Packet

The Remove Training Line Packet removes the training lines. It begins with the packet header 8, and ends with a newline character.

Plane of Cut Packet

The plane of cut packet begins with the packet header 9, the scissor pose, and contains four triplets of x, y, z points in the septum plane coordinate system, given in clockwise order. The packet ends in a newline

, a timestamp, and a flag for whether the line of cut is valid (whether it intersects the septum plane). All values are tab spaced, and the packet ends in a newline.

The scissor rotation is given in three Euler angles, it's the scissor rotation relative to the septum plane, the scissor translation is three doubles, the line of cut start is a vector with three doubles, the line of cut end is a vector with three doubles, the timestamp is a long since unix epoch, and valid is a flag that is 0 if the line of cut is not in the septum plane, and 1 otherwise.

The packet starts with the packet prefix, and contains all fields in the order given, they are tab-spaced, and it ends in a newline.

Training Plane Packet

The training plane packet begins with the packet header A, and contains four triplets of x, y, z points in the septum plane coordinate system, given in clockwise order. The packet ends in a newline