OpenIGTLink Protocol Version 2

Release 0.00

Junichi Tokuda[1], Nobuhiko Hata[1]

June 15, 2011

[1]Department of Radiology, Brigham and Women's Hospital and Harvard Medical School, Boston, MA

Abstract

In this paper, we review the technical details of new features in OpenIGTLink version 2. OpenIGTLink is a free open-source network communication protocol designed for data exchange among devices and software. Since the first release of version 1.0 protocol along with a reference C/C++ library in 2008, the number of developers and users, who use OpenIGTLink in their research projects, have been growing among convincing some commercial companies to implement research interface based on OpenIGTLink their commercial products. However, the message types initially defined for research prototypes does not cover some of functionality in commercial products, thus the protocol needed to be requires refinement in the protocol. The goal of the extension in version 2 is to support such new demands emerged after the release of the initial version, while keeping the protocol as simple as possible. The version 2 protocol has been implemented as an extension of version 1; it is backward compatible in the sense that messages generated by OpenIGTLink version 2 compliant software can be received by version 1 compliant software.

Contents

1 Introduction

Device/software communication is essential in today's IGT. It is quite common to locate surgical tools relative to the patient's body by using position tracking systems with optical (1), electromagnetic (2), or ultrasonic (3,4) sensors, or to acquire images using real-time ultrasound, computed tomography, or magnetic resonance imaging (MRI). Those sensor and image data are often transferred to surgical navigation software for visualization enabling the physicians to know the accurate target lesion, or monitor the progress of treatment noninvasively. In addition, the use of robotic assistance is becoming reality, requiring additional communication between the navigation software and robotic hardware. Therefore, major task for researchers and developers in the field of IGT is to design a system consisting of multiple devices and software, and integrate them by establishing communication among those components. Currently, many vendors provide proprietary application programming interface (API) that allows external software/hardware communicating with their products. However, lack of standardized interface raises serious constraints in the system integration; the developers are forced to spend considerable time to understanding and implementing device-specific interfaces; the interfaces from different vendors are not compatible; proprietary interface often requires special contract that comes with non-disclosure agreement and/or expensive license fee.

To address the abovementioned issue, we launched a project to provide a standardized mechanism to connect software/hardware through the network for IGT applications. The core engineering effort is development of an open network communication protocol, OpenIGTLink (5). The protocol is defined as a set of simple binary message formats, which are necessary to transfer various types of data exchanged among the software/hardware used in IGT. The protocol is provided as a document as well as a free open-source reference C/C++ library with example programs. The idea behind our open-source strategy is to maximize the number of products that have OpenIGTLink compliant interfaces by providing a freely-available, simple, extensible and reliable protocol that attracts a wide range of developers including academic researchers and industrial developers to the community. Since the first release of version 1.0 protocol along with a reference C/C++ library and interface in medical image processing and visualization software, 3D Slicer, in 2008, the number of developers and users, who use OpenIGTLink in their research projects, have been growing especially among academic researchers convincing some commercial companies to implement research interface based on OpenIGTLink their commercial products. However, the message types initially defined for research prototypes does not cover some of functionality in commercial products, thus the protocol needed to be requires refinement in the protocol.

Since Sumer Project Week 2010 of NIH-funded National Alliance for Medical Image Computing, we have been working on extension of the protocol, which will be released as OpenIGTLlink protocol version 2. The goal of the extension is to support the new demands emerged after the release of the initial version, while we keep the protocol as simple as possible. The version 2 protocol has been implemented as an extension of version 1; it is backward compatible in the sense that messages generated by OpenIGTLink version 2 compliant software can be received by version 1 compliant software. The new protocol defines a set of new message types with new querying scheme. In this paper, we review the technical details of new features.

2 Overview of the Protocol

The OpenIGTLink protocol itself does not include mechanisms to establish and manage a session. A message, the minimum data unit of this protocol, contains all information necessary for interpretation by the receiver. The message begins with a 58-byte header section, which is common to all types of data, followed by a body section. The format of the body section varies by data type, specified in the header section. Since any compatible receiver can interpret the header section, which contains the size and data type of the body, every receiver can gracefully handle any message, even those with unknown data type. Therefore, this two-section structure allows developers to define their own data type while maintaining compatibility with other software that cannot interpret their user-defined data types. This simple message mechanism eases the development of OpenIGTLink interfaces and improves compatibility.

3 New Querying Scheme in Version 2

The OpenIGTLink protocol defines special message types to support querying data, which allows requesting the receiver to send data back. Those message types have message name with special prefixes. In addition to “GET_” prefix to request a single message from the receiver, version 2 also support query messages to control streaming data, which is a series of messages in the same data type e.g. real-time instrument tracking and real-time imaging. A client can start and stop data streaming from a server, by issuing a query message with a device type starting with "STT_" and "STP_" respectively. If data requested by a "STT_" message is not available, the server returns a message with a device type string starting with "RTS_" prefix. A STP_ message is also acknowledged by a "RTS_" message. This is useful to start and stop position tracking or real-time imaging remotely from the client. For example, once the server receives "STT_TDATA" message from the client, it start sending "TDATA" messages to the client. The server keep sending "TDATA" messages until it receives "STP_TDATA" from the client.

4 New Message Types

The available message types are listed in Table 1. In this section, we review the message types newly added in protocol version2.

QTRANS. The QTRANS data type is used to transfer position and orientation information. The data are a combination of 3-dimensional vector for the position and quaternion for the orientation. Although equivalent position and orientation can be described with the TRANSFORM data type, the QTRANS data type has the advantage of smaller data size (19%). It is therefore more suitable for pushing high frame-rate data from tracking devices.

IMGMETA. The IMGMETA message is used to transfer image meta information which are not available in IMAGE message type, such as patient name, medical record number, modality etc. An IMGMETA message can contain meta data for multiple images. This message type may be used to obtain a list of images available in the remote system, such as image database or commercial image-guided surgery (IGS) system.

LBMETA. To retreive voxel objects or a label map, GET_IMAGE / IMAGE can be used. But the client should be able to get a list of available structures.

COLORT. COLORT is used to transfer a color table. This allows sharing label image data with the same color assignment.

POINT. The POINT message type is designed to transfer information about fiducials, which are often used in surgical planning and navigation in the image-guided therapy.

TRAJ. The TRAJ message type support to transfer information about 3D trajectory, which is often used in surgical planning and guidance in image-guided therapy.

TDATA. The TDATA message type is intended for transferring 3D positions of surgical tools, markers etc. Those positions are often measured by optical, electromagnetic or other type of 3D position sensor continuously and transferred as series of messages. Since it is important for software that receives TDATA to control data flow, STT_TDATA query data type has interval field to control the frame rate of consecutive messages.

QTDATA. The QTDATA message type is intended for transferring 3D positions of surgical tools, markers etc. Its role is almost identical to TDATA, except that QTDATA describes orientation by using quaternion.

SENSOR. SENSOR is a message type, which is used to transfer sensor reading, 3-axis position, velocity, acceleration, angle, angle velocity and angle acceleration. The message format is intended for manipulator control and various types of sensors. The SENSOR message can handle a part of unites defined in The International System of Unites (SI) in its 8-byte (or 64-bit) field. The field is designed to specifiy a unit consisting of SI-prefix (e.g. milli, micro, kilo etc...) and combination of SI-base and/or SI-derived unites.

NDARRAY. NDARRAY is a data type, which is designed to transfer N-dimensional numerical array.

BIND. Bind message format is designed to bind any OpenIGTLink messages into a single message. Messages bound in a BIND message are called 'child messages.' The BIND message format is useful, when one needs to care about synchronization of multiple messages (e.g. multi-channel sensors), or sending associative array (pairs of key string and value). The bind message format consists of the bind header section, which contains types and sizes of child messages, the child message name table section, and the child message body section.

POLYDATA. POLYDATA is used to transfer 3D polygonal data. The message format is designed based on the POLY DATA format defined in VTK file format and equivalent to VTK's vtkPolyData class.

5 Discussion

In this paper, we review the technical detail of the new features in OpenIGTLink protocol version 2. The extended querying mechanism and the new message types are defined based on the analysis of current use cases and existing research network interface in existing surgical navigation system. The version 2 protocol is backward compatible; the all new message types has the same header format, and interpreted as user-defined messages by version 1 complaint software; the new querying mechanism is implemented as a scheme of exchanging messages, which follows the OpenIGTLink message format.

The extension of the OpenIGTLink protocol is an important milestone in our effort to provide a standardized network communication protocol that focuses on IGT applications. Since the initial release of OpenIGTLink protocol, it has been gradually gaining users and developers among academic researchers, who developed prototype system for IGT applications based on open-source technologies, resulting a number of popular open-source software packages that support OpenIGTLink interfaces, e.g. Image-Guided Surgery Toolkit (IGSTK) (6), 3D Slicer (7), CISST Library (8). This growing number of users and developers has also been motivating some commercial venders to provide OpenIGTLink-compliant interfaces in their proprietary products, which are essential to our effort for standardization. The extension of OpenIGTLink protocol provides simple but wide range of data transfer, reducing amount of engineering resource required to adapt their products to the standard. Increasing number of commercial products that support OpenIGTLink protocol will attracts more users and developers in the field, resulting the expanded research community as well as increasing opportunity for the venders to attract more users to their commercial products.

Reference

1. Bucholz RD, Smith KR, Henderson J. Intraoperative localization using a three-dimensional optical digitizer. SPIE Volume 1894; 1993. p 312-322.

2. Birkfellner W, Watzinger F, Wanschitz F, Ewers R, Bergmann H. Calibration of tracking systems in a surgical environment. IEEE transactions on medical imaging 1998;17(5):737-742.

3. Barnett GH, Kormos DW, Steiner CP, Weisenberger J. Intraoperative localization using an armless, frameless stereotactic wand. Technical note. Journal of neurosurgery 1993;78(3):510-514.

4. Roberts DW, Strohbehn JW, Hatch JF, Murray W, Kettenberger H. A frameless stereotaxic integration of computerized tomographic imaging and the operating microscope. Journal of neurosurgery 1986;65(4):545-549.

5. Tokuda J, Fischer GS, Papademetris X, et al. OpenIGTLink: an open network protocol for image-guided therapy environment. Int J Med Robot 2009;5(4):423-434.

6. Enquobahrie A, Cheng P, Gary K, et al. The image-guided surgery toolkit IGSTK: an open source C++ software toolkit. J Digit Imaging 2007;20 Suppl 1:21-33.

7. Gering DT, Nabavi A, Kikinis R, et al. An integrated visualization system for surgical planning and guidance using image fusion and an open MR. J Magn Reson Imaging 2001;13(6):967-975.

8. Deguet A, Kumar R, Taylor RH, Kazanzides P. The cisst libraries for computer assisted intervention systems. The MIDAS J -- Syst Archit Comput Assist Interv 2008.