

Slice based pipeline protocol

Jan-Willem Buurlage

1 Introduction

This document describes a communication protocol for a distributed tomographic reconstruction pipeline in which parameters used in the reconstruction can be changed in real-time, taking effect on the running reconstruction immediately.

The protocol is based on the reconstruction of individual slices, typically ortho-planes, and is useful for situations where the projection data is too big to reconstruct completely in real-time. The slices are shown together in a 3D interface, and get updated when e.g.:

- new (projection) data is available
- more iterations for iterative solvers have been applied
- higher resolution reconstructions are available
- ...

The position and orientation of the *active slices* can be changed, and this is communicated back to the reconstruction cluster, which for future updates will then reconstruct the new slices.

2 Pipeline overview

Initially we will focus on the following ‘pipeline topology’. In a typical setting, the different stages of the pipeline are as follows:

- data source \rightarrow preprocessing \rightarrow reconstruction \leftrightarrow controller \leftrightarrow visualizer

We will ignore the left part of this pipeline for now, and assume that in the ‘reconstruction’ the proper data is available (or streamed in at real time). We will define the *two-way communication* between the controller, the visualizer and the reconstructor. Their roles are roughly the following:

- The **reconstructor** manages the incoming (preprocessed) projection data, and registers itself with the controller to create a **scene**. Associated to a scene are the **active slices** and their *orientations*. Note that a

reconstructor can actually be a cluster of computers, but they are all associated to a single *scene* (dataset or real-time measurement).

- The **controller** manages the list of scenes, and corresponding slices, and communicates with the reconstructor(s) about changes in preferred resolution, orientation and so on. There are two reasons that this controller exists; 1) it separates the visualizer from the reconstructor which simplifies the implementation of a visualizer, and 2) it easily allows for the existence of multiple scenes which is useful for running reconstructions with different parameters or when using the pipeline as a plotting utility.
- The job of the **visualizers** is to visualize any given scene. It can also communicate using a simple protocol with the controller for changes in (reconstruction) parameters. This is mostly because it can provide a convenient user interface for changing these parameters.

The reason for looking at (ortho)slice-based reconstruction is largely because of data management. To prevent bottlenecks, we do not want to send large parts of data around, nor do we want to reconstruct large volumes. This is why the controller *never* gets to see the raw data, which lives only on the reconstruction node(s).

2.1 Packages

The communication between nodes happens in a using standardized *network packets* that contain data, commands, or parameters. For the precise content of specific packages see Section 6.

3 Extensions

Possible extensions for the future include:

- Low resolution 3d reconstruction that is shown when moving the slices.

4 Implementation details

We currently plan to use *ZeroMQ* because of its flexibility with respect to programming languages and communication strategies, and its speed. An alternative could be *protocol buffers* which have been developed by Google.

5 Software examples

- Make a package:

```
// C++  
auto packet = MakeScenePacket("Test scene #1");  
packet.send();  
  
# Python  
packet = MakeScenePacket("Test scene #1");  
packet.send();
```

- Send a package:
- Receive a package / reply:

6 Appendix: Package definitions

```
// Usage: register a scene with the controller  
// Reply: a `scene_id`  
packet MakeScene:  
    string name  
    int dimension  
    real[dimension] volume_geometry  
  
packet UpdateImage:  
    int slice_id
```