Laboratory for Percutaneous Surgery

PLUS overview

(Public software Library for UltraSound imaging research)

Andras Lasso, Tamas Heffter, Csaba Pinter, Tamas Ungi,
Thomas Kuiran Chen, Alexis Boucharin,
and Gabor Fichtinger

Queen's University

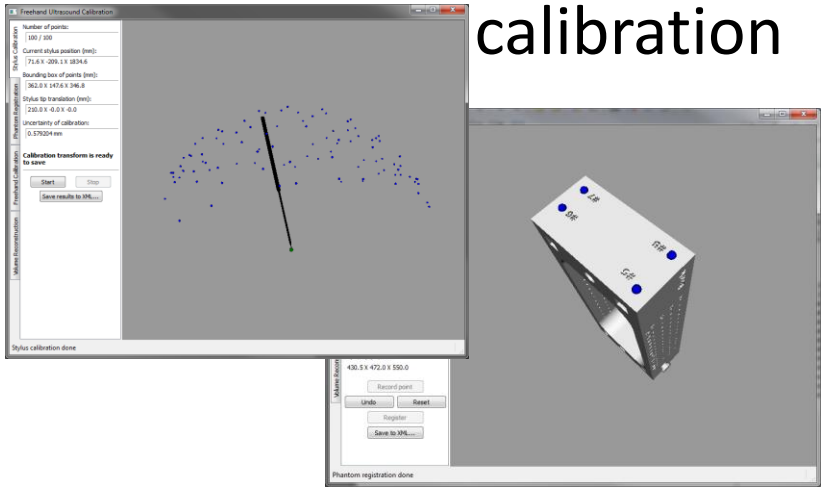
Introduction

- PLUS – **P**ublic software **L**ibrary for **U**ltra**S**ound imaging research
- Goal: facilitate rapid prototyping of ultrasound-guided intervention systems for translational clinical research
- Acquire, process, transfer *synchronized* ultrasound image and position tracking data
- History: implementation is based on two SynchroGrab versions
  - QueensOpenIGTLibs in Queen's repository
    Last commit: October 7, 2008 (Revision: 30)
    svn+ssh://image.cs.queensu.ca/image/svn/QueensOpenIGTLibs/trunk/VTKLibs/Ultrasound
  - 4D Ultrasound module in NAMIC sandbox repository
    Last commit: August 16, 2009 (Revision: 4993)
    http://svn.na-mic.org/NAMICSandBox/ trunk/4DUltrasound-WithGating/
- Fixed several issues, completely reworked – funded by SparKit
- Open-source (since October 2011): BSD license, no strings attached

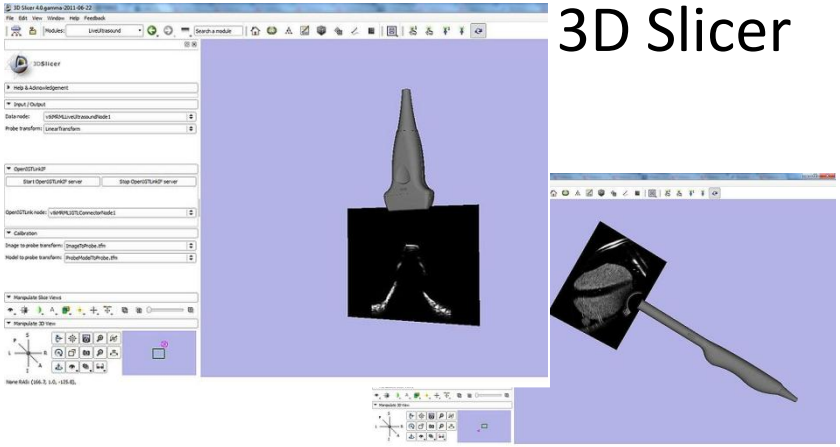
Highlights

- Spatial and temporal calibration: compute image plane to marker transform, using double-N calibration phantom
- Tracked ultrasound capturing: B-mode and RF
- Volume reconstruction
- Live data transfer through OpenIGTLink to 3D Slicer and other compatible apps
- Diagnostics for image and tracking data acquisition
- Support of multiple hardware devices
- Complete solution: documentation, tutorials, CAD models, sample data, simulators, automatic tests, examples

Spatial calibration



Live data in 3D Slicer



Supported hardware

Position trackers

- Ascension EM tracker
- NDI Certus optical tracker
- NDI Polaris optical tracker, NDI Aurora EM tracker (WIP)
- Claron MicronTracker optical tracker
- Brachy steppers (CMS Accuseed, Burdette Medical systems, CIVCO)
- Simulator

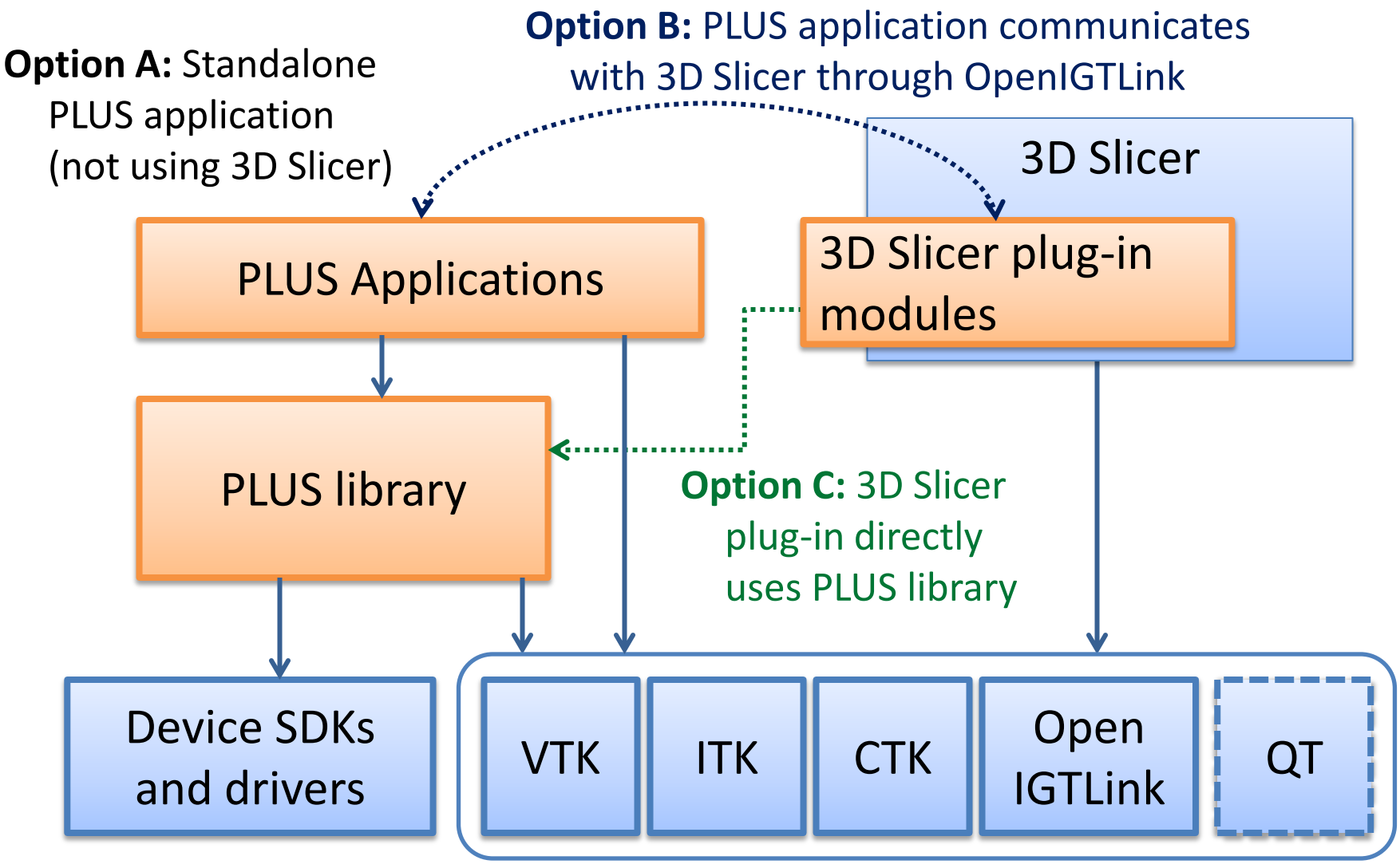
Imaging devices

- Ultrasonix B-mode & RF (digital)
- ImagingControl framegrabber (analog)
- Simulator

Single configuration XML file

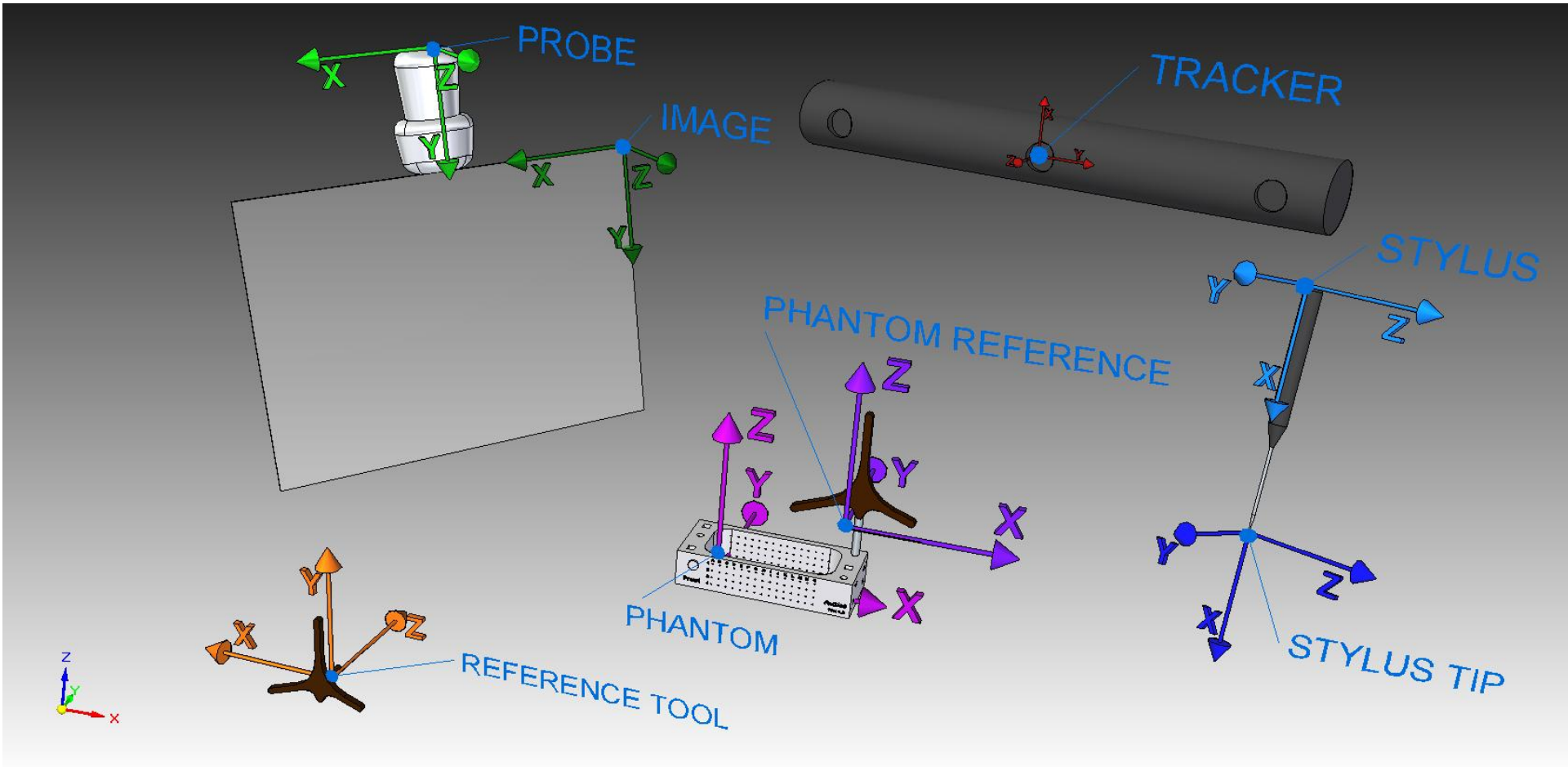
- Defines input parameters for all components of Plus, grouped as "**Device set**" for easy selection in GUI applications.
- Main sections
  - USDataCollection
    - Tracker (e.g. Certus, Ascension) + Tools (probe, reference, stylus)
    - Image source (e.g. SonixVideo, Frame grabber)
    - Synchronization
  - USCalibration
  - Calibration phantom definition
  - Volume reconstruction
    - Output spacing
    - Slice clipping
    - Probe calibration
  - …

Software platform



Spatial calibration

Goal: determine the PROBE to IMAGE transform

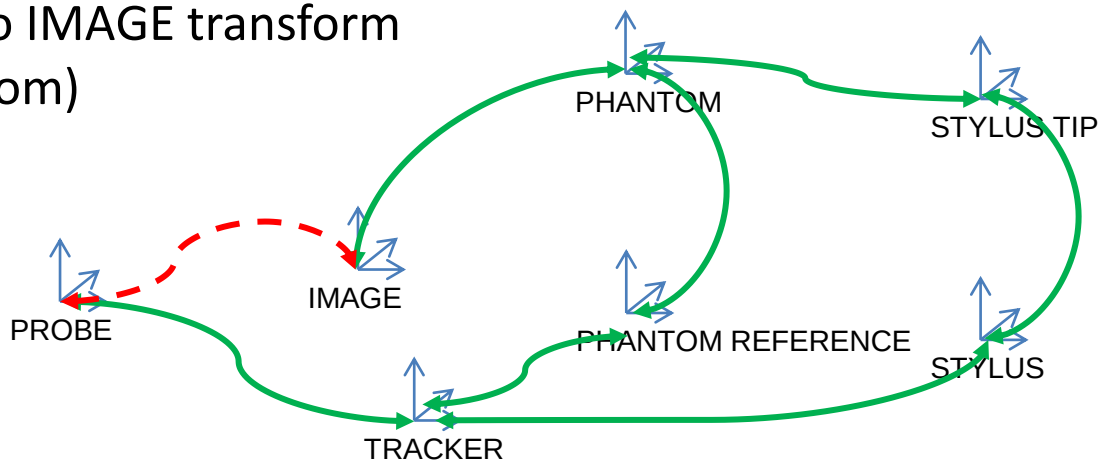


Spatial calibration

Goal: determine the PROBE to IMAGE transform

Spatial calibration Steps

1. Temporal calibration (measure delay between imaging and position tracking)
2. Determine STYLUS to STYLUS TIP transform (pivot calibration)
3. Determine PHANTOM to PHANTOM REFERENCE transform (landmark registration)
4. Determine the IMAGE to PHANTOM transform (fiducial line segmentation)
5. Determine PROBE to IMAGE transform (using N-wire phantom)



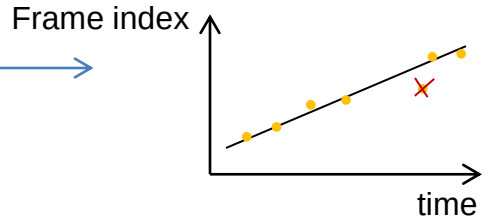
Spatial calibration

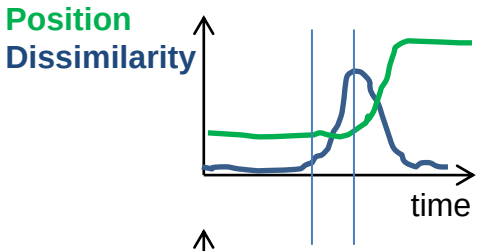
Tutorials with all data, models, tricks

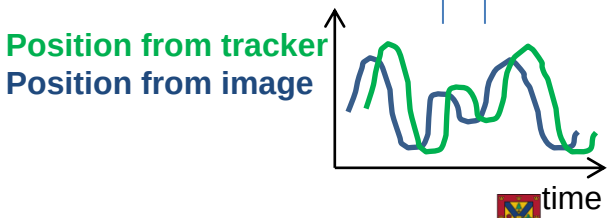
- Performing tracked ultrasound probe calibration using fCal
- How to build an fCal calibration phantom

Temporal calibration

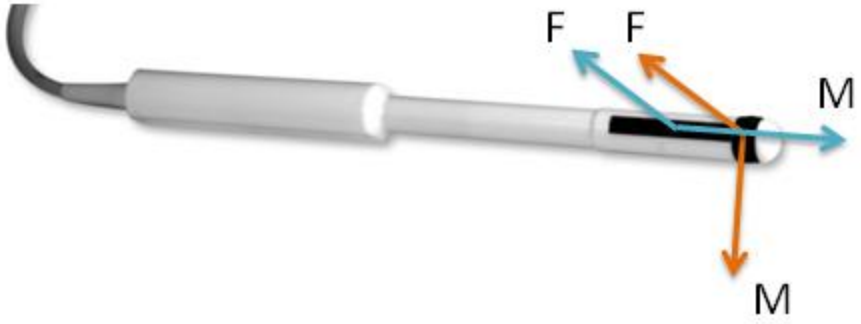
- Previous implementations were really bad
- “Accurate” software timing: multi-media timers
- Verification with periodic motion with robot
- Filtering: simple linear model
- Limit maximum speed
- Lag estimation methods:
  - Change-detection based
  - Correlation-based (WIP)

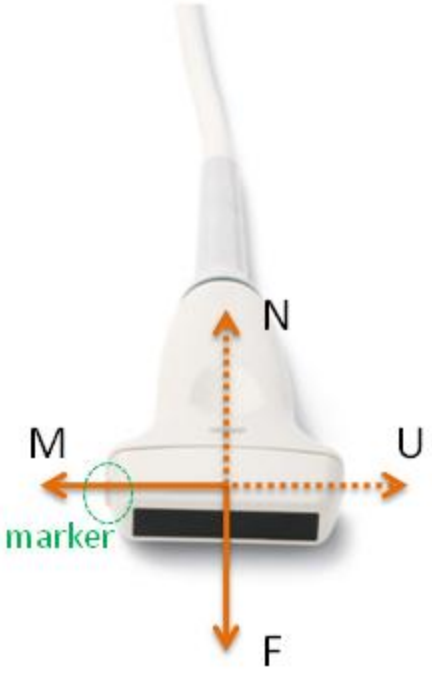






Ultrasound image orientation





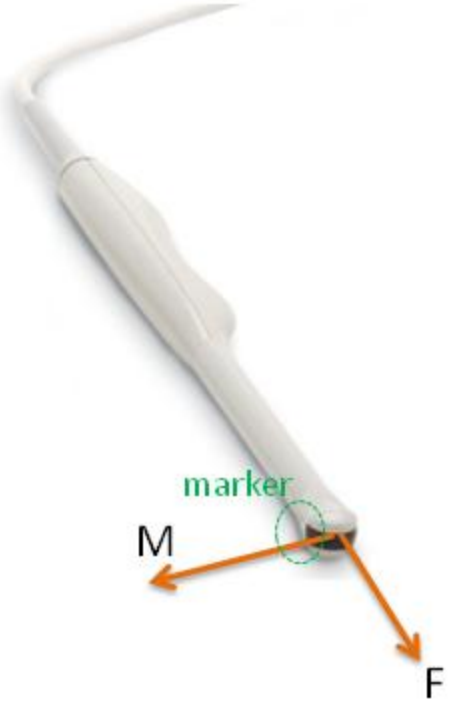
Transducer axes:

M = marked

U = unmarked = -marked

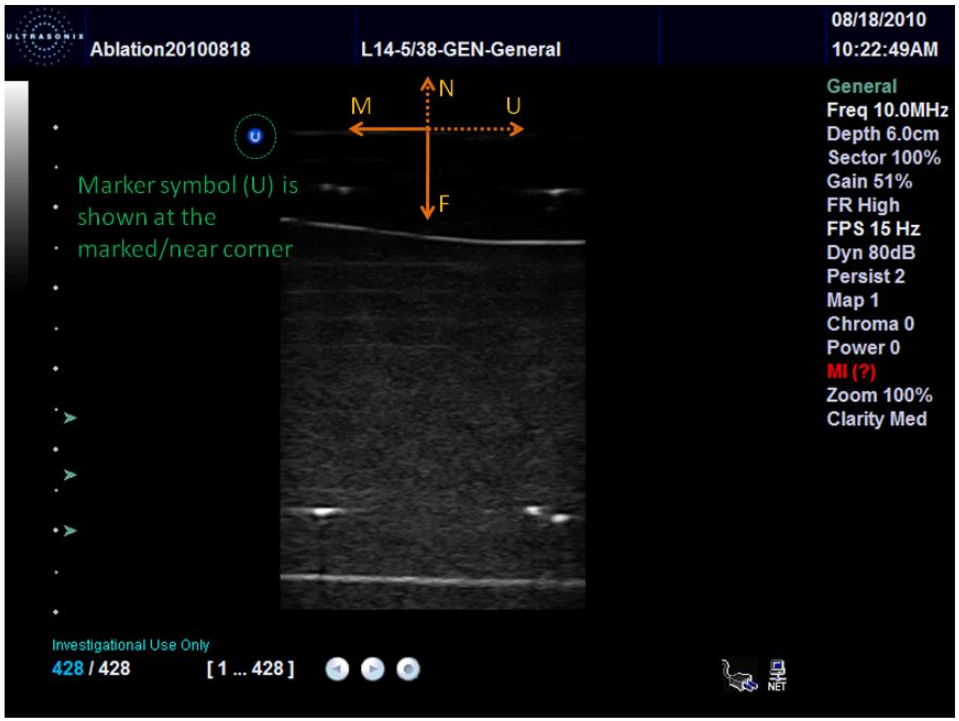
F = far

N = near = -far



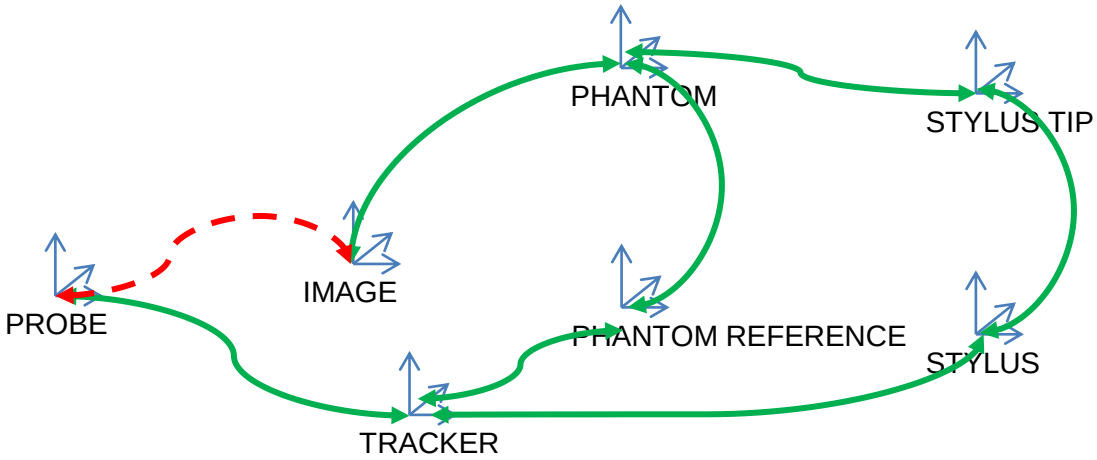
Ultrasound image orientation

- Plus stores all images in MF
- By default, SonixTouch displays UF for linear probe, UN for endocavity probe



Transform repository

- Specify known transforms between coordinate systems
  - SetTransform(“Probe”, “Image”, probeToImageTransformMatrix)
  - SetTransform(“Probe”, “Tracker”, probeToTrackerTransformMatrix)
- Retrieve transform between any two coordinate systems
  - GetTransform(“Image”, “Tracker”, imageToTrackerTransformMatrix)

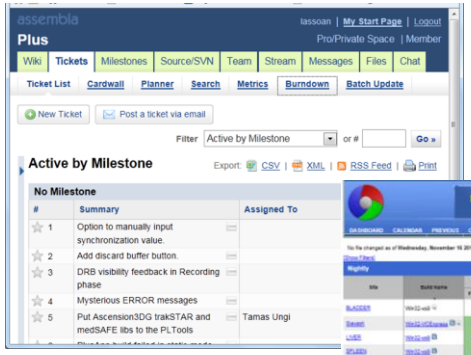


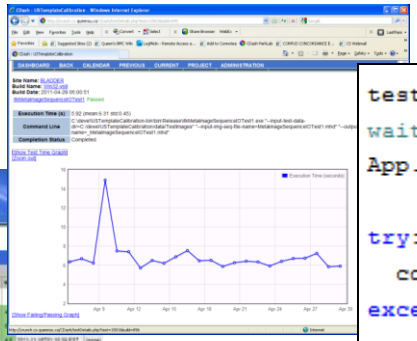
Sequence metafile (.mha)

- Extension to the *Meta IO* standard file format
- Slices readable by many existing applications
- Extra information for tracking/reconstruction
  - Frame number
  - Unfiltered and filtered timestamp
  - Probe and Reference tracking transforms
  - …
- Used by all applications of Plus

Software process

- Source control, tickets, releases, messaging (www.assembla.com/spaces/plus)
- Standardized build environment (CMake automatically downloads and configures all required software components
- Documentation: wiki, doxygen
- Automatic tests: CTest, CDash, Sikuli



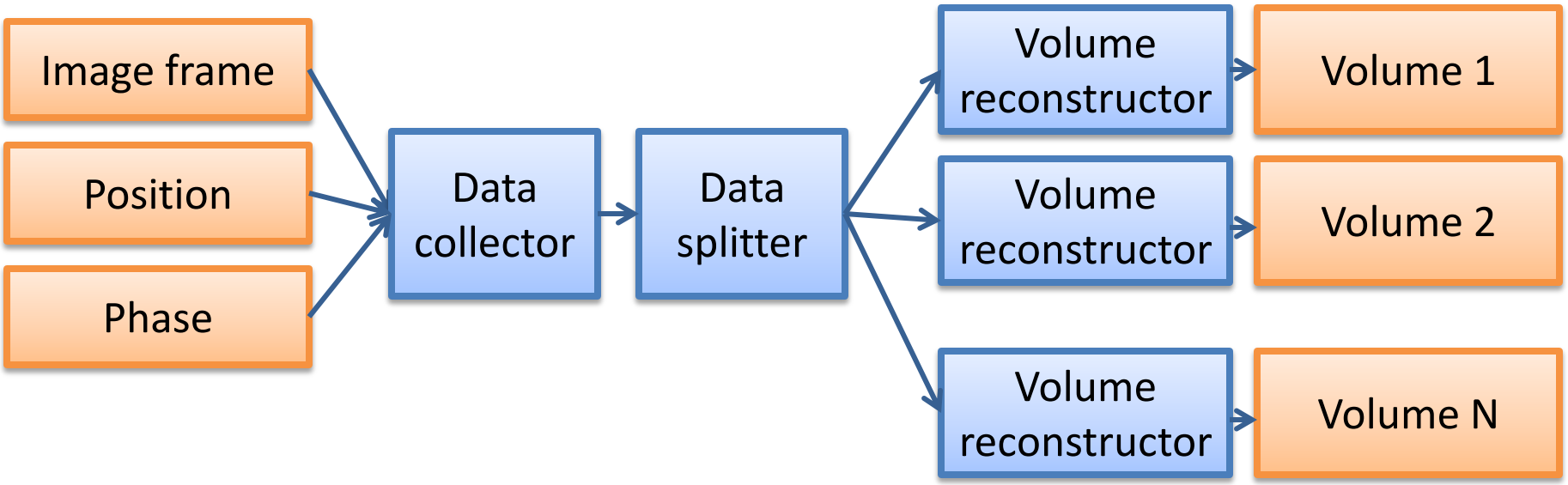


```
testApp = App.open(exe)
wait(10) # Wait for the application to initialize (else
App.focus(appTitle)

try:
  connectButton = wait( Connect , 60)
except FindFailed:
  print "[ERROR] Application did not start!"
  captureScreenAndExit()

# Get the region of the segmentation parameter dialog w
applicationTopLeft = connectButton.getTopLeft()
```

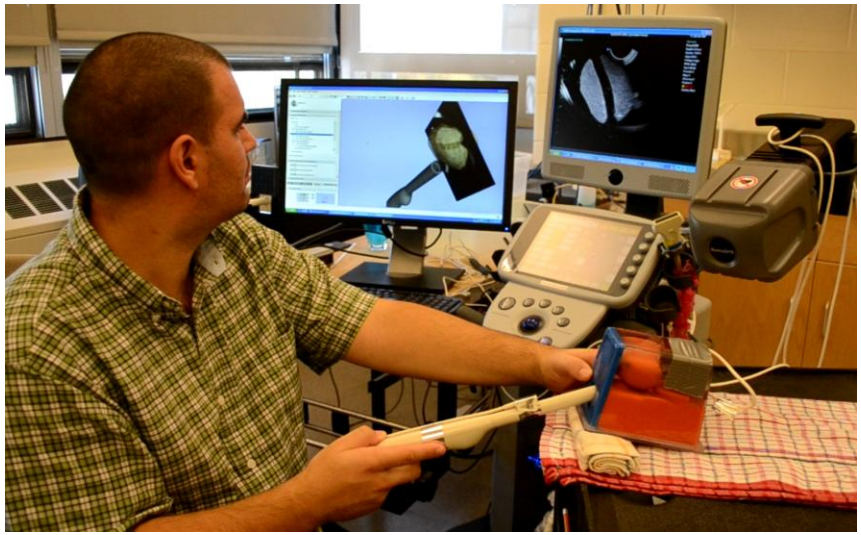
4D reconstruction



- Phase: ECG (some A/D converter devices supported, but it's untested), transducer angle (supported for Ultrasonix), …
- Data collector, volume reconstructor is ready
- Data splitter has not been implemented

Summary

- Tracked ultrasound
  - Calibration (spatial, temporal)
  - Diagnostics
  - Acquisition
  - Processing
  - Live transfer (OpenIGTLink)



- Extensive hardware support
- Hardware abstraction
- Software process (high software quality, limited efforts)
- Open-source, free to use

https://www.assembla.com/spaces/plus/

http://perk.cs.queensu.ca

Appendix

Coordinate systems

- ProbeCalibrationMatrix

Probe->Image or Image->Probe?

- ProbeTransform

Probe->Tracker or Tracker->Probe?
or Probe->Reference or Reference->Probe?

- ImageToProbeTransform 
- ProbeToTrackerTransform 
- (CoordinateSystem1)To(CoordinateSystem1)Transform 

Queen's University

Spatial calibration

Generic description of the phantom

```
</USCalibration>
    <!-- PHANTOM -->
- <PhantomDefinition version="1.1">
        <!-- Supported types are: Double-N, U-Shaped-N -->
    <Description Name="fCAL" Type="Double-N" Institution="Queen's University PerkLab" Version="1.0"/>
        <!-- ModelToPhantomTransform - transforming model into phantom coordinate system for proper visualization -->
    <Model File="FCal_1.0.stl" ModelToPhantomTransform=" 1 0 0 -15.0 0 1 0 10.0 0 0 1 -5.0 0 0 0 1"/>
  - <Geometry>
            <!-- N wire definitions -->
      - <Pattern Type="NWire">
            <Wire Name="1:E3_e3" EndPointBack="20.0 40.0 5.0" EndPointFront="20.0 0.0 5.0"/>
            <Wire Name="2:F3_j3" EndPointBack="45.0 40.0 5.0" EndPointFront="25.0 0.0 5.0"/>
            <Wire Name="3:K3_k3" EndPointBack="50.0 40.0 5.0" EndPointFront="50.0 0.0 5.0"/>
        </Pattern>
      - <Pattern Type="NWire">
            <Wire Name="4:E4_e4" EndPointBack="20.0 40.0 0.0" EndPointFront="20.0 0.0 0.0"/>
            <Wire Name="5:J4_f4" EndPointBack="25.0 40.0 0.0" EndPointFront="45.0 0.0 0.0"/>
            <Wire Name="6:K4_k4" EndPointBack="50.0 40.0 0.0" EndPointFront="50.0 0.0 0.0"/>
        </Pattern>
            <!-- Landmark list for registration -->
      - <Landmarks>
            <Landmark Name="#1" Position="95.0 5.0 15.0"/>
            <Landmark Name="#2" Position="95.0 40.0 15.0"/>
            <Landmark Name="#3" Position="95.0 40.0 0.0"/>
            <Landmark Name="#4" Position="95.0 0.0 0.0"/>
            <Landmark Name="#5" Position="-25.0 40.0 15.0"/>
            <Landmark Name="#6" Position="-25.0 0.0 10.0"/>
            <Landmark Name="#7" Position="-25.0 0.0 0.0"/>
            <Landmark Name="#8" Position="-25.0 40.0 0.0"/>
        </Landmarks>
    </Geometry>
</PhantomDefinition>
```