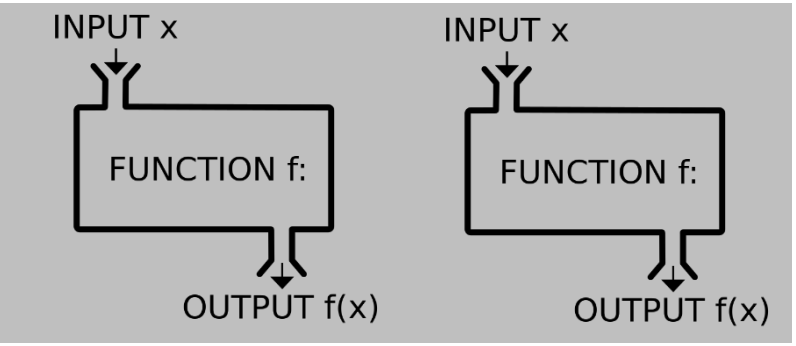
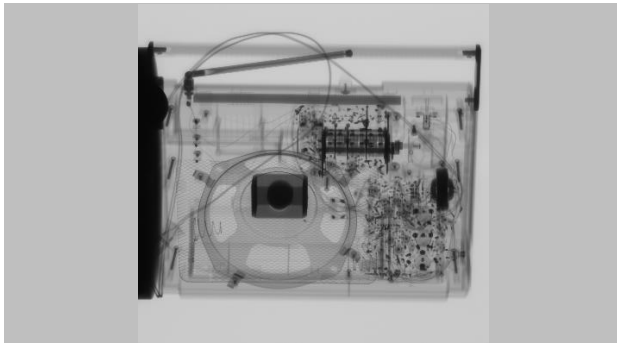
Exceptional service in the national interest

Sandia National Laboratories





A Generalizable Radiography Algorithm Test Environment for NDE Applications

Andrew C. Wantuch, Jaxon M. Gittinger, Ismael Perez, Edward S. Jimenez

Sandia National Laboratories, Software Systems R&D

U.S. DEPARTMENT OF ENERGY | NNSA National Nuclear Security Administration

Sandia National Laboratories is a multi-mission laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND NO. 2011-XXXXP

So What Who Cares?

Sandia National Laboratories

- **Topic**
  - ATR Algorithm Development
- **Problem 1**: Need for a fast, *open architecture* ATR test environment that doesn't require access to systems to evaluate different types of 3rd party ATR algorithms
  - **Solution**: Provide 3rd party ATR algorithm developers with a way to develop ATR algorithms using pre-existing scans
    - Inspired by earlier efforts at ALERT for automated scoring (TO4)
- **Problem 2**: No standardized, open architecture method of comparing ATR algorithms
  - **Solution**: Provide a standardized way to benchmark algorithms
- **So What?**
  - Reduced barrier to entry for algorithm developers
  - Potentially speed up certification processes
  - Build confidence for adoption of 3rd party contributions

Open Threat Assessment Platform (OTAP)

Sandia National Laboratories

- Develop and demonstrate an open architecture baggage screening prototype
- What is "Open"?
  - Standardized across vendors
  - Modular
  - Plug-and-play
- Allow 3rd Party Development of:
  - Hardware
  - Software
  - Algorithms
- Partner with security technology manufacturers

Test Environment Objectives

Sandia National Laboratories

- Evaluate the performance of algorithms developed by third-parties using a common image database
  - Standardized metrics
  - Standardized timing
  - Programming language agnostic
- Be simple and easy for algorithm developers to use
  - No complex emulators
    - Emulators often need every component of the screening system implemented
  - Be highly flexible to support all conceivable algorithms
    - Variable input/output methods
    - Nontraditional approaches
- Enable iterative algorithm development

Design

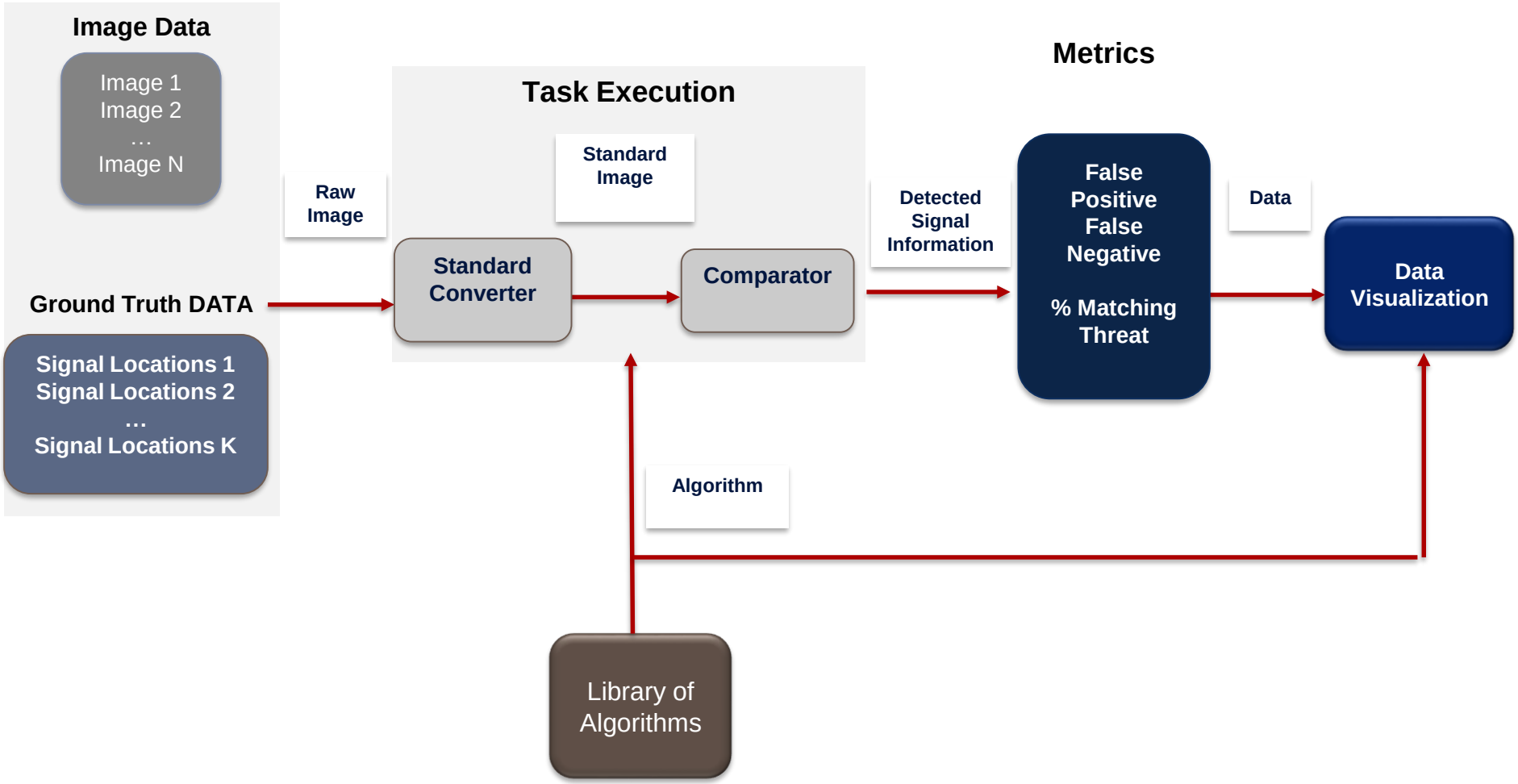


Image Data
Image 1
Image 2
…
Image N
Ground Truth DATA
Signal Locations 1
Signal Locations 2
…
Signal Locations K
Raw Image
Task Execution
Standard Image
Standard Converter
Comparator
Detected Signal Information
Metrics
False Positive
False Negative
% Matching Threat
Data
Data Visualization
Algorithm
Library of Algorithms

Example

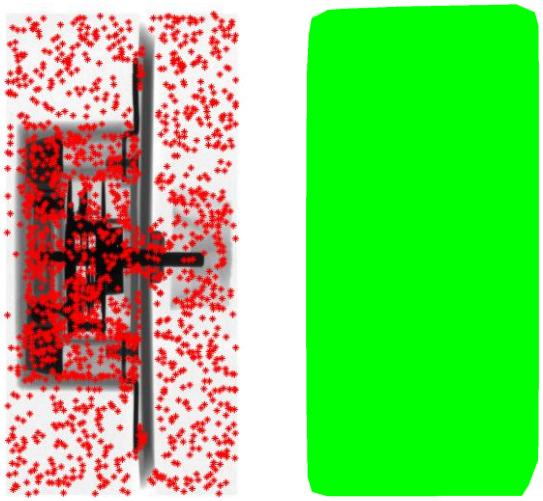
- Algorithms: SIFT and SURF
  - Popular computer vision algorithms
  - Identify features in images such as corners and changes in contrast
  - Only feature locations used for this example
- Database: Radiographs of various COTS components
- Ground Truth: Features extracted by Matlab SIFT

SIFT Features

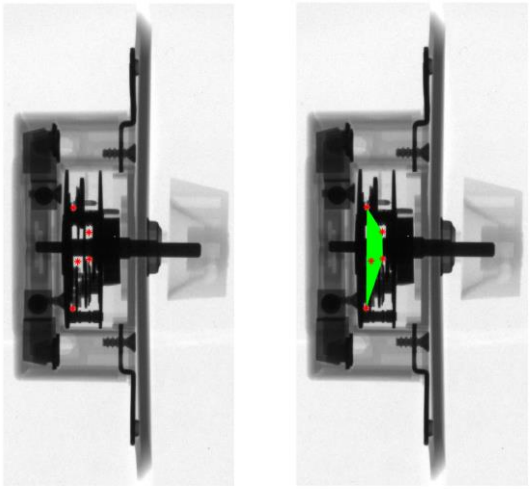
SURF Features

Example – Continued

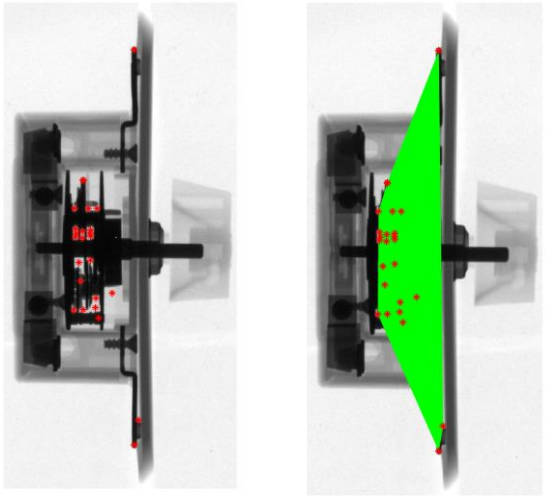
Ground Truth: 1948 points



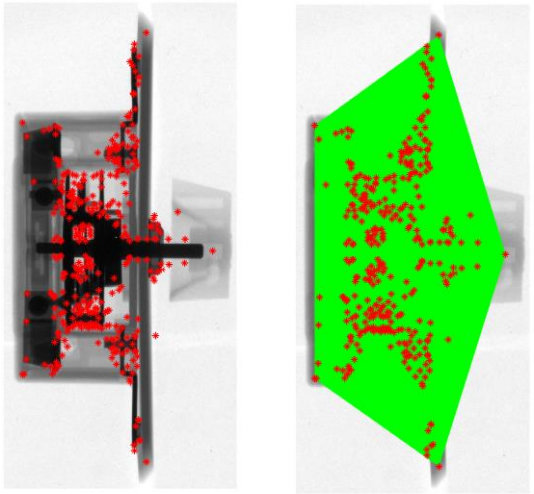
Algorithm 1: Use 5 Points



Algorithm 2: Use 25 Points



Algorithm 3: Use All Points (349)



Results

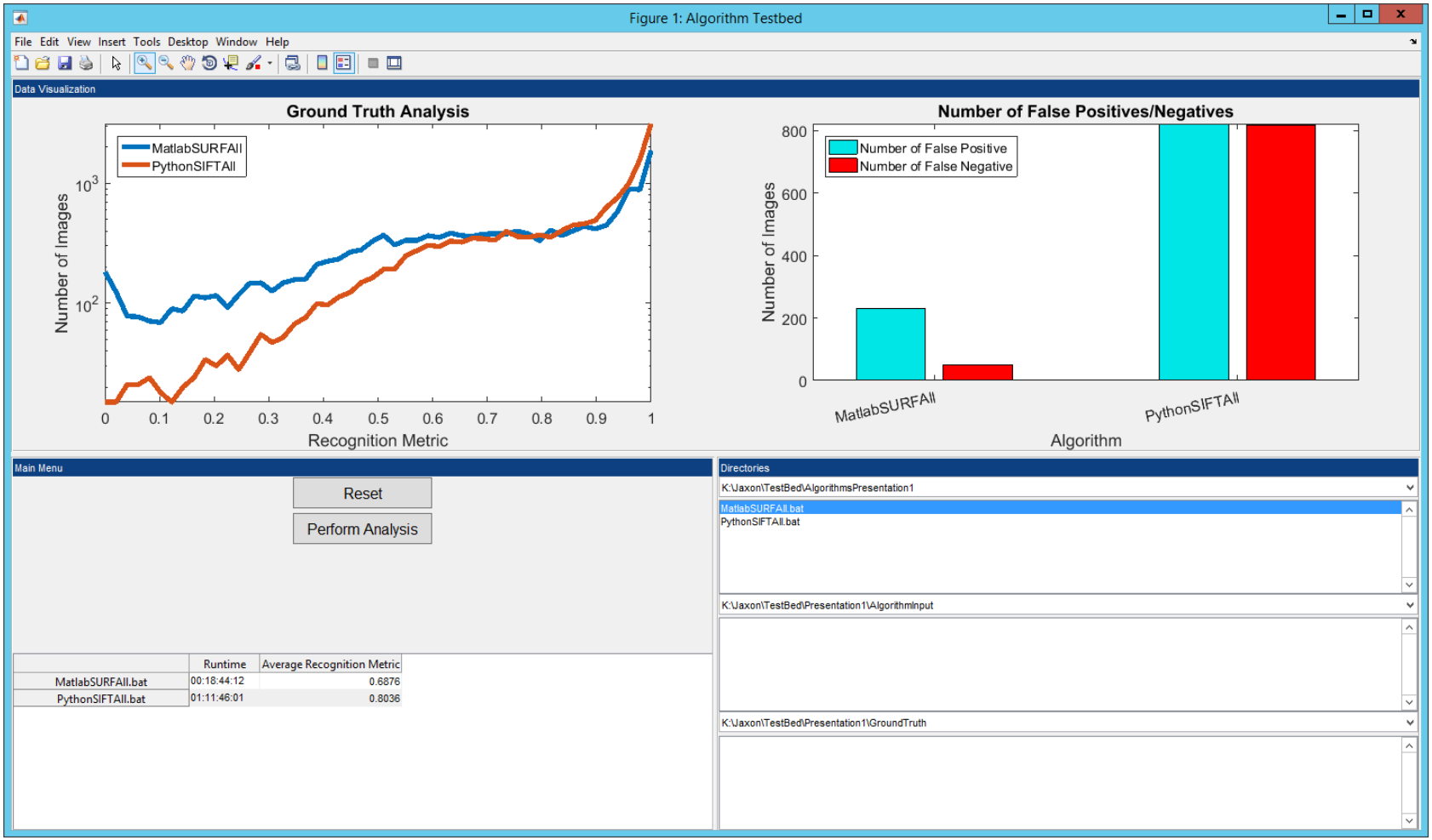
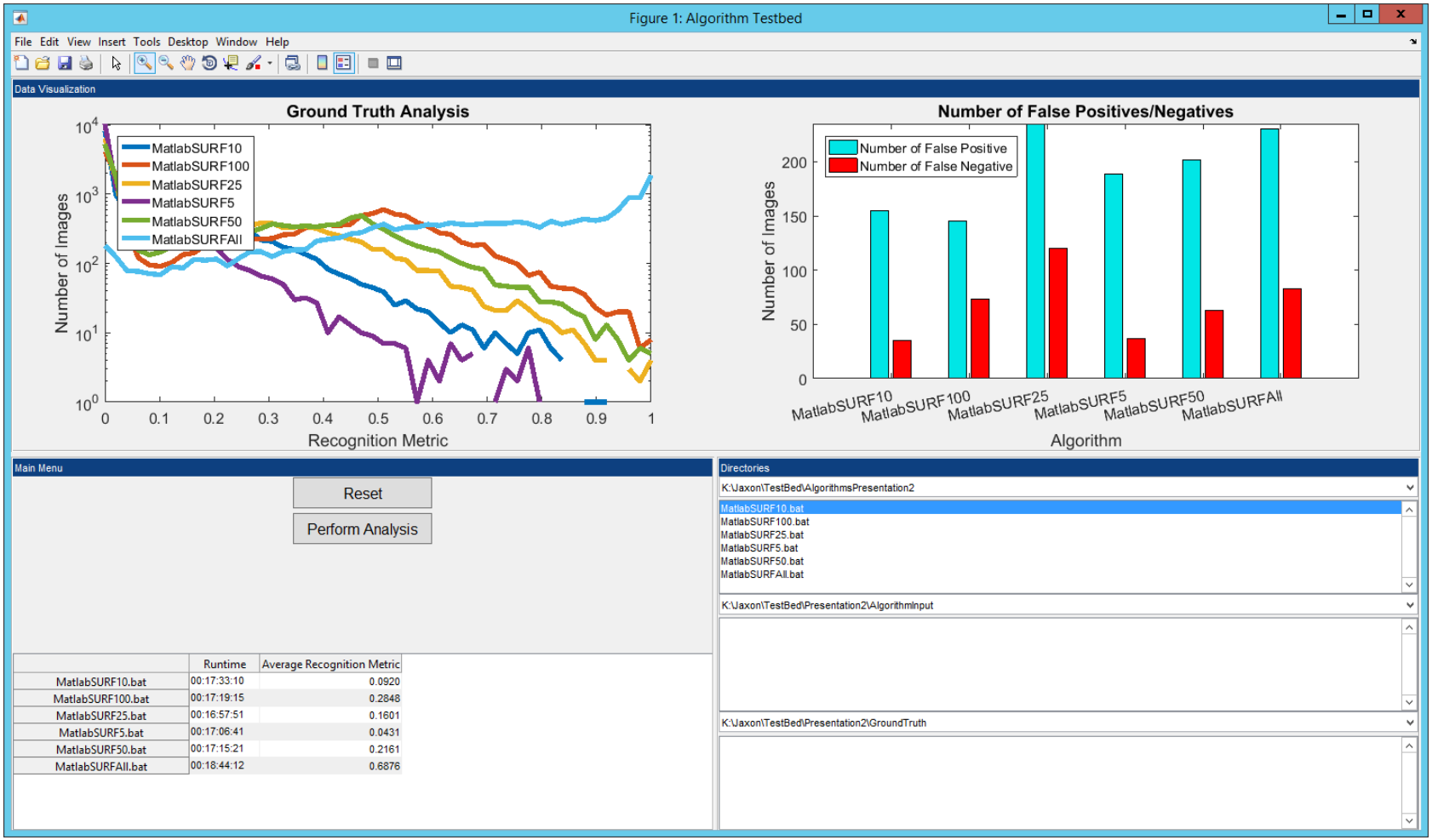


Figure 1: Algorithm Testbed
File Edit View Insert Tools Desktop Window Help
Data Visualization
Ground Truth Analysis
MatlabSURFAll
PythonSIFTAll
Number of Images
Recognition Metric
Number of False Positives/Negatives
Number of False Positive
Number of False Negative
Number of Images
MatlabSURFAll
PythonSIFTAll
Algorithm
Main Menu
Reset
Perform Analysis
Runtime
Average Recognition Metric
MatlabSURFAll.bat
00:18:44:12
0.6876
PythonSIFTAll.bat
01:11:46:01
0.8036
Directories
K:\Jaxon\TestBed\AlgorithmsPresentation1
MatlabSURFAll.bat
PythonSIFTAll.bat
K:\Jaxon\TestBed\Presentation1\AlgorithmInput
K:\Jaxon\TestBed\Presentation1\GroundTruth

Results



	Runtime	Average Recognition Metric
MatlabSURF10.bat	00:17:33:10	0.0920
MatlabSURF100.bat	00:17:19:15	0.2848
MatlabSURF25.bat	00:16:57:51	0.1601
MatlabSURF5.bat	00:17:06:41	0.0431
MatlabSURF50.bat	00:17:15:21	0.2161
MatlabSURFAll.bat	00:18:44:12	0.6876

Conclusions

Sandia National Laboratories

- Implemented a functional Prototype in Matlab
  - Likely supports algorithms written in any programming language
    - Tested with Python and Matlab
  - Generates standardized metrics for algorithms
  - Compares multiple algorithms or multiple versions of the same algorithm
  - Helps with rapid and iterative development of new algorithms with lower barrier to entry

Next Steps

Sandia National Laboratories

- Support DICOS files as input
- Support CT datasets
- Determine method of deployment
  - Web app?
  - Distribute to 3rd parties?
  - Keep in-house at TSA/SNL?
- Investigate security concerns
  - How can we securely execute someone else's executables?
- Work with vendors to provide what they want/need

Questions?

Sandia National Laboratories

Backup Slides

Sandia National Laboratories

OTAP Enables Plug-and-Play