Evaluation and Exploration of Next Generation Systems for Applicability and Performance

Volodymyr Kindratenko

Guochun Shi

Activities

- Lossless data compression
- Check sums
- Database appliance
- Image characterization for document comparison

Lossless data compression

- Variety of lossless data compression techniques
  - Dictionary-based encoders (Lempel-Ziv family of algorithms) replace input strings with pointers to a dictionary data structure updated by the encoder.
  - Variable-length entropy encoders, such as Huffman and arithmetic coding, assign prefix codes to each input symbol based on its statistical frequency of occurrence.
  - Run-length coders store repeated values as a single instance and a count.
- Best lossless data compression techniques work by using probabilistic models in which predictions are coupled to an arithmetic coding algorithm.

Lossless data compression

- To date, no serious prior work on general-purpose lossless data compression on GPUs exists
  - The closest is 2009 students work from Stanford University
- But there is some good prior work for floating-point data compression
  - Idea is based on the suppression of leading zeros in the residuals between the input values and their predicted values
  - This technique however is not suitable for general-purpose use
  - And its performance is limited by the PCIe bus bandwidth (~6 GB/sec)

Lossless data compression

- GPU implementation of a general-purpose lossless data compressor suffers due to
  - Serial nature of the algorithms
    - Parallelization via data partitioning is possible, but not sufficient to utilize massive parallelization of GPUs
    - Control instructions abundant nature of the algorithms
    - Irregular data access patterns
  - PCIe bandwidth
    - Currently peak at 8 GB/s for PCIe gen. 2 x16
    - Compared to 10-17 GB/s for DDR3 memory modules
- We have not yet found a general-purpose lossless data compression algorithm amenable for an efficient GPU implementation

Check sums

- NIST is currently working with the cryptographic community to develop a new cryptographic hash algorithm, “SHA-3”, to become a new standard.
- Currently five final candidates are considered.
  - BLAKE
  - Grøstl
  - JH
  - Keccak
  - Skein
- GPU implementation of these algorithms is of great interest, with significant work already in existence.

Check sums

- Since cryptographic hash function usually contains only bits operations, GPUs can achieve very high performance given enough parallelism.
  - First-round MD6 has been implemented on NVIDIA GPU [Mohan’10].
  - Implementation of most of the second round candidate algorithms on NVIDIA GPU is done too [Bos’10].
- However, since the original message data is in host memory and has to be copied to GPU through PCIe, the best performance cannot exceed the PCIe bandwidth (~6 GB/s).
  - Bos’s best GPU implementation of BLAKE-32 and BMW-256, works at 4.6 GB/s.

Database appliance

- We still do not have the hardware from XtremeData
  - Currently its delivery is scheduled for March 28
- We do have a database from NARA and example queries. The database, however, is too small to efficiently stress the database engine.
  - We will need to design a test procedure for this.

Image characterization

- We have integrated a GPU-based implementation of SIFT algorithm [Wu'10] with **doc2learn** pdf file comparison software as a replacement for the probability density function based image comparison algorithm.
  - This allows to compare images embedded in pdf files based on their actual content rather than on a color histogram.
- We also have integrated a GPU-based implementation of SIFT algorithm with **Versus** framework developed by the Image Spatial Data Analysis Group at NCSA.

Introduction to SiftGPU

- A GPU implement to Scale-Invariant Feature Transformation(SIFT) algorithm from Univ. of North Carolina at Chapel Hill
  - Extract Features from an image
  - Compare features from different images and find matching ones
- Both feature extraction and comparison are done in GPU

SiftGPU with Versus

ImageObject
BufferedImage
PDF Document
ImageJ
…

Histogram E.
Texture E.
Array E.
…

Histogram F.
Texture F.
Array F.
…

Pearson
Subtraction
Chi Squared
…

Diagram courtesy to Luigi Mariti

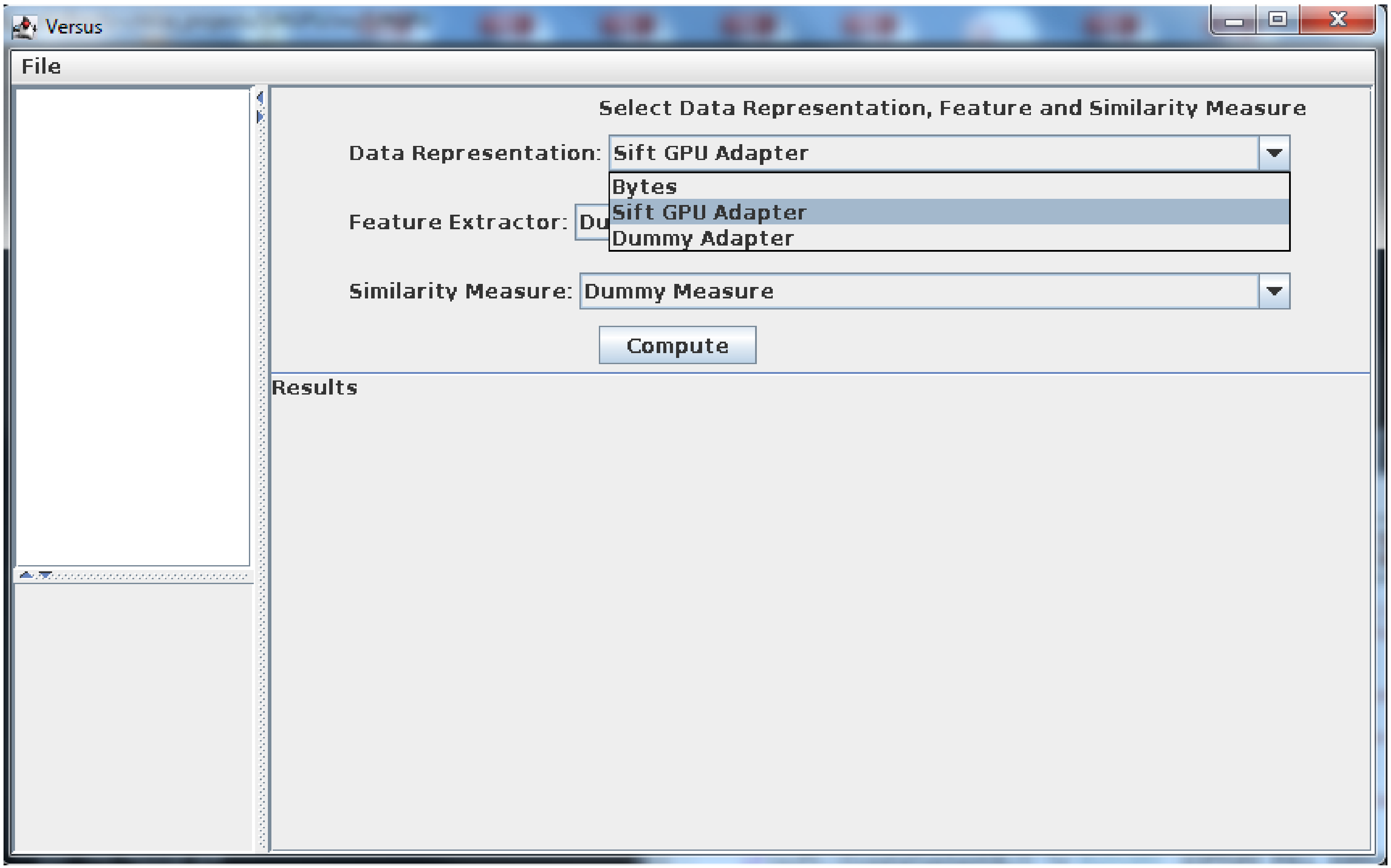
- Implemented the Adapter/Extractor/Measure/Feature classes in the Versus framework
- The final similarity value is computed as

$$s = \frac{n}{\max(N1, N2)}$$

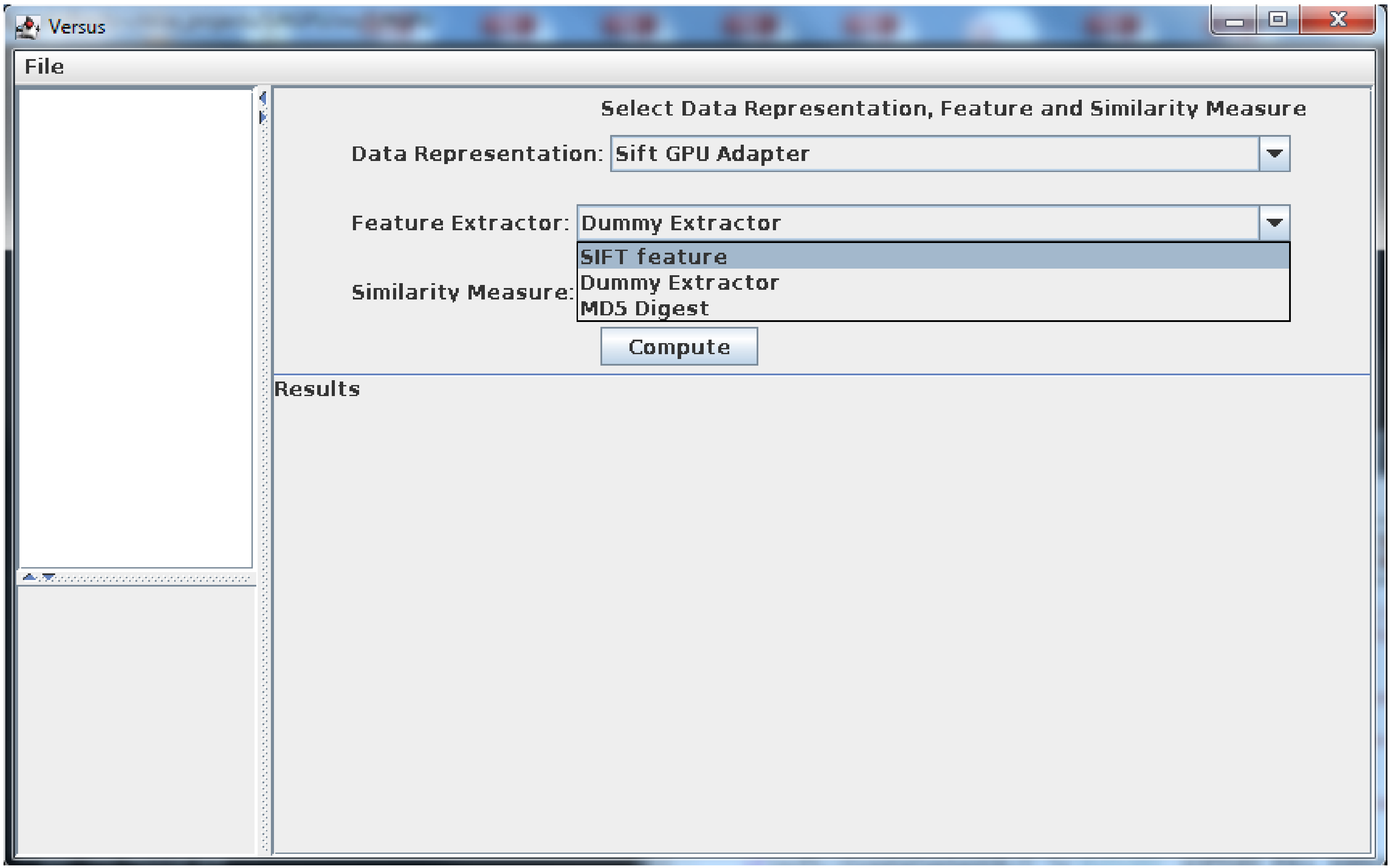
where

N1: # of features in image 1
N2: # of images in image 2
n: # of matched features

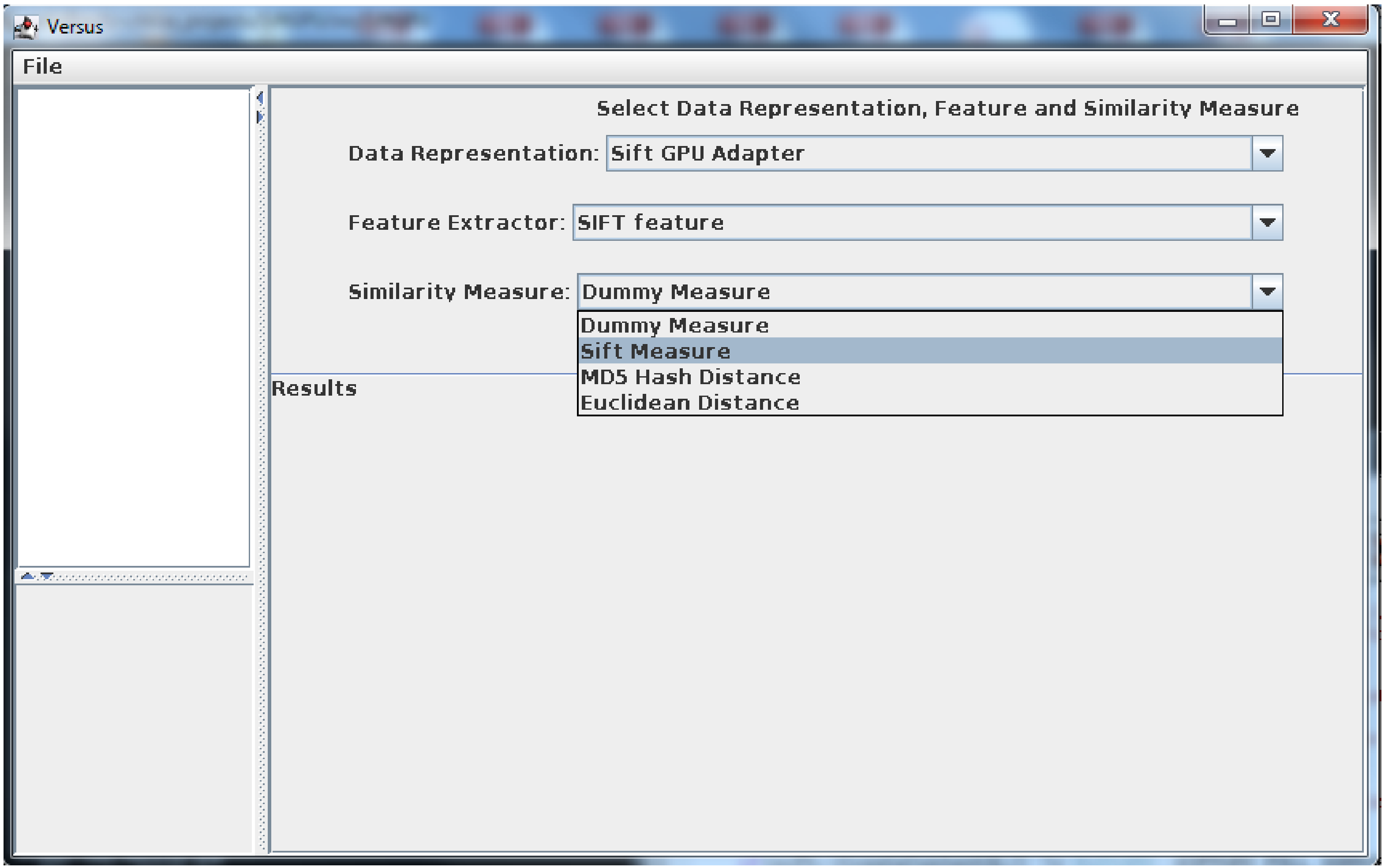
Versus example



Versus example

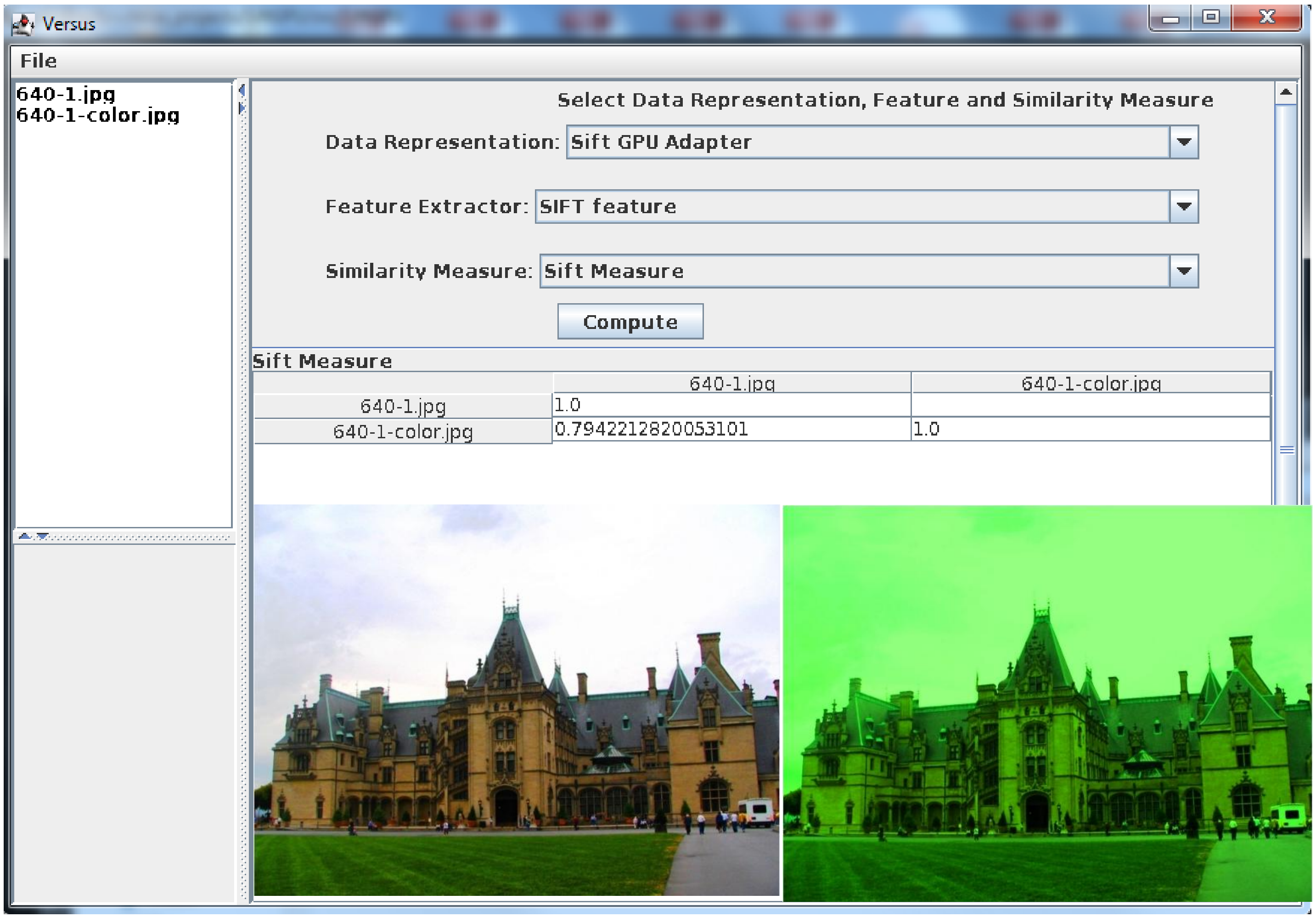


Versus example



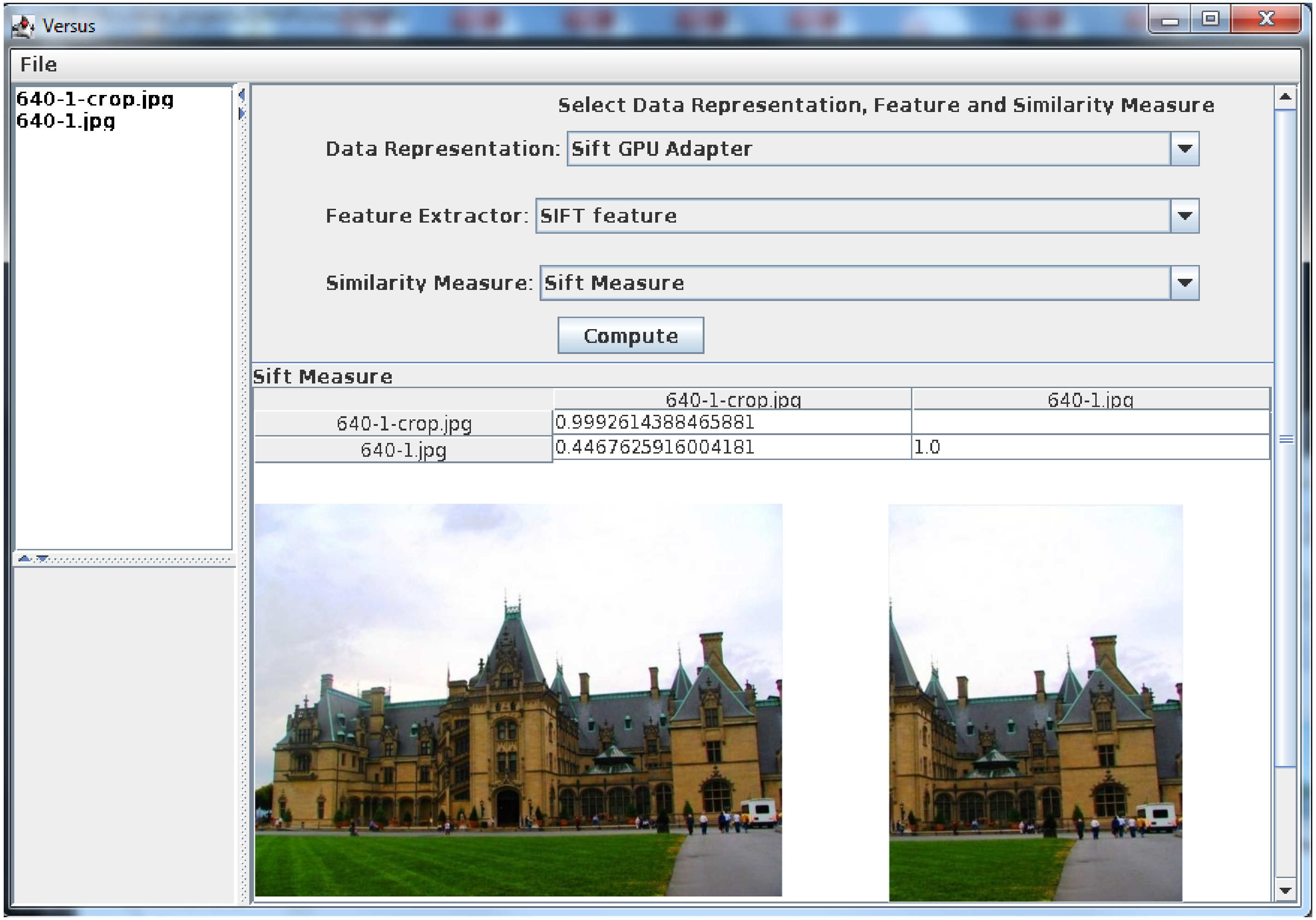
Versus example

- Original image vs the color-changed one
- Similarity : 0.79



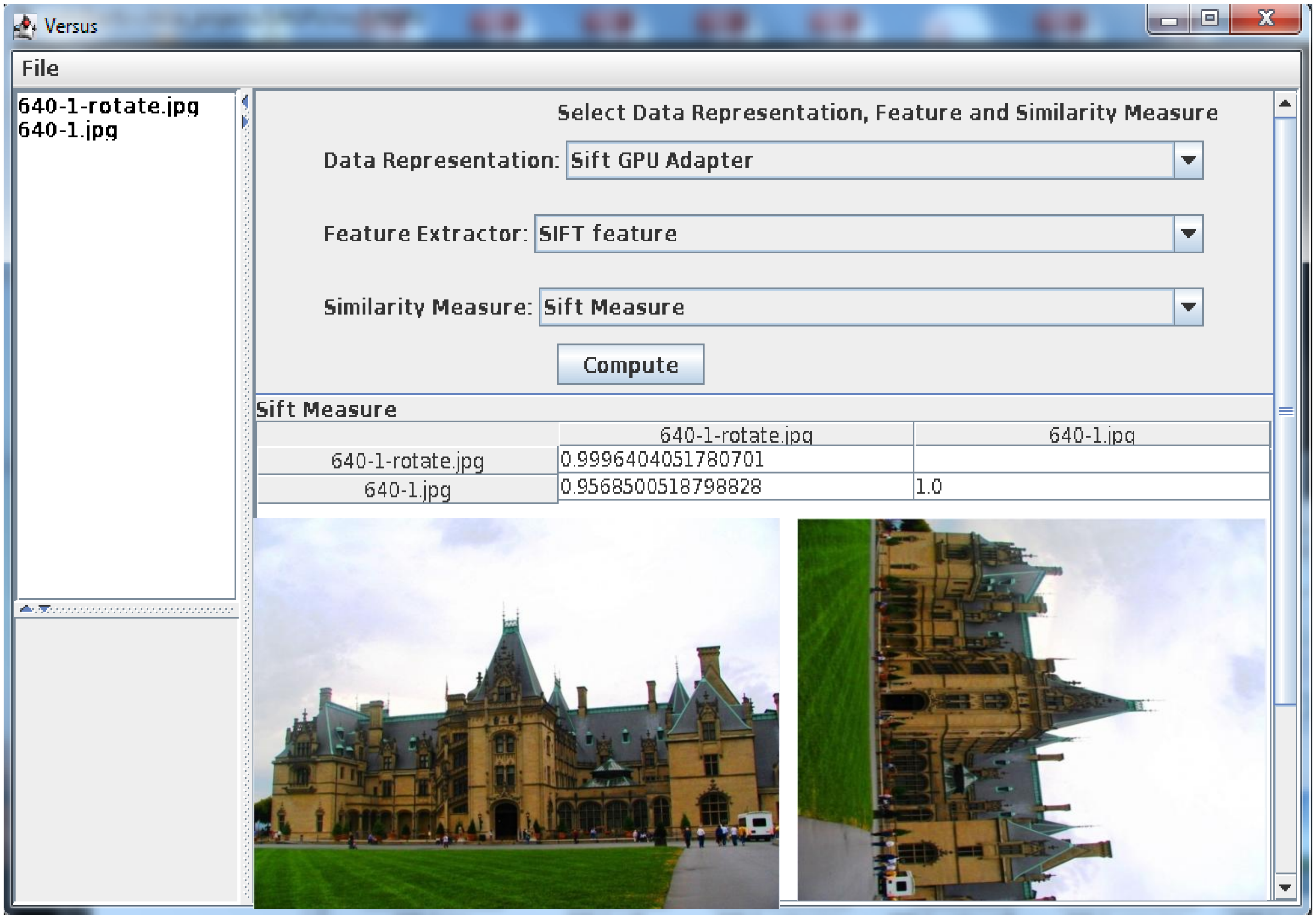
Versus example

- Original image vs half of original image
- Similarity : 0.45



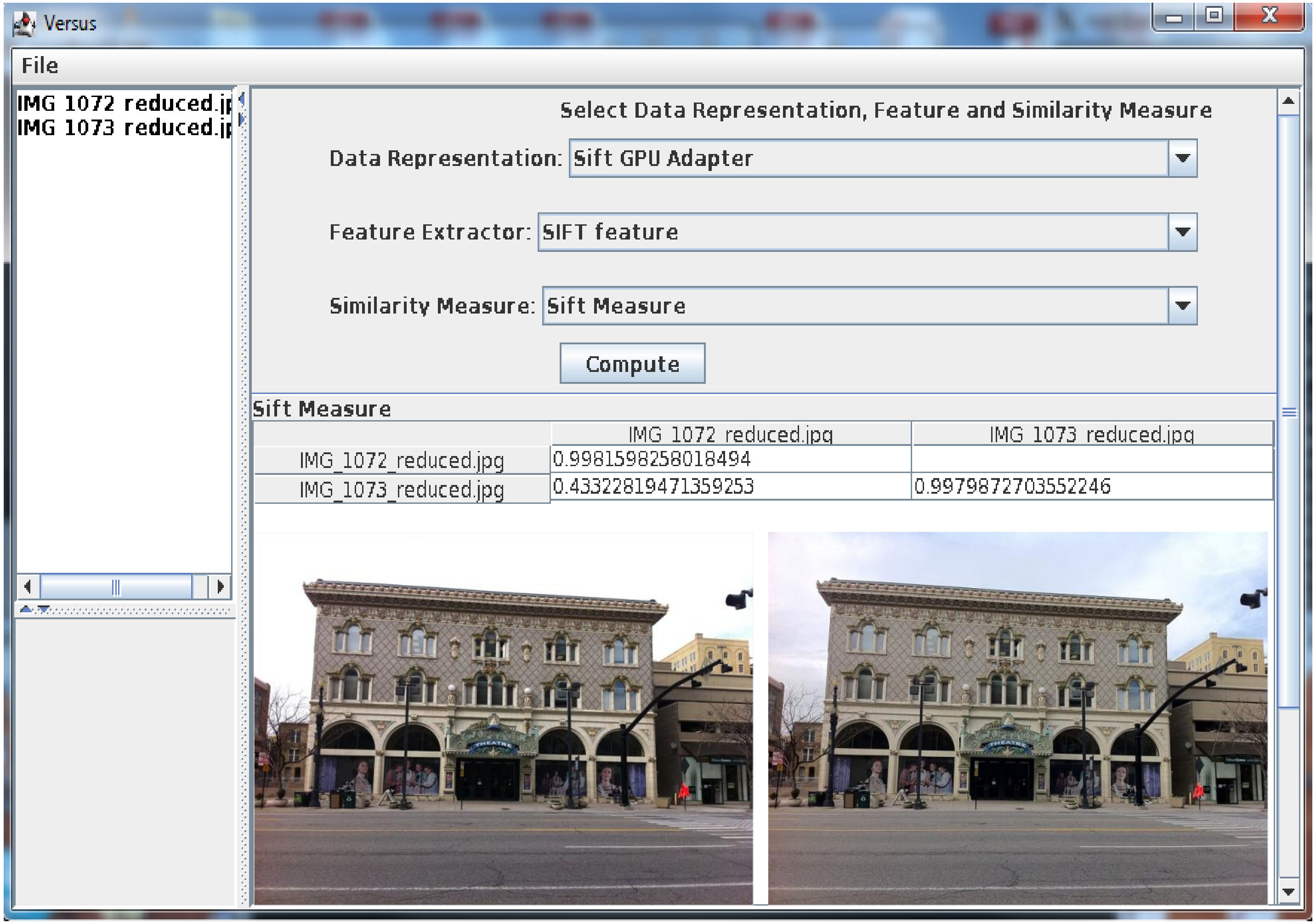
Versus example

- Original image vs the rotation of original image
- Similarity : 0.96



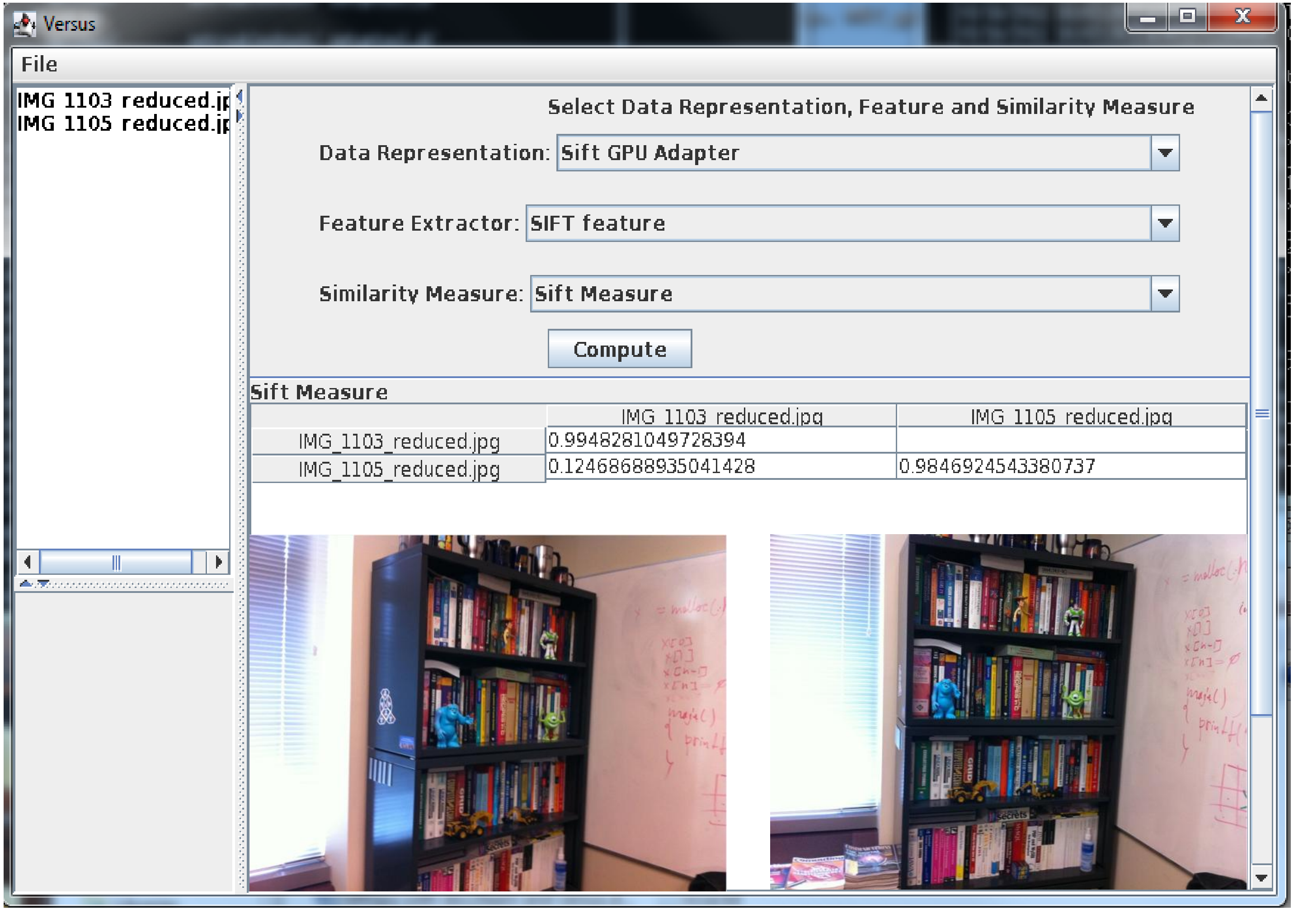
Versus example

- Two images take from the same scene but different light exposure time
- Similarity : 0.43



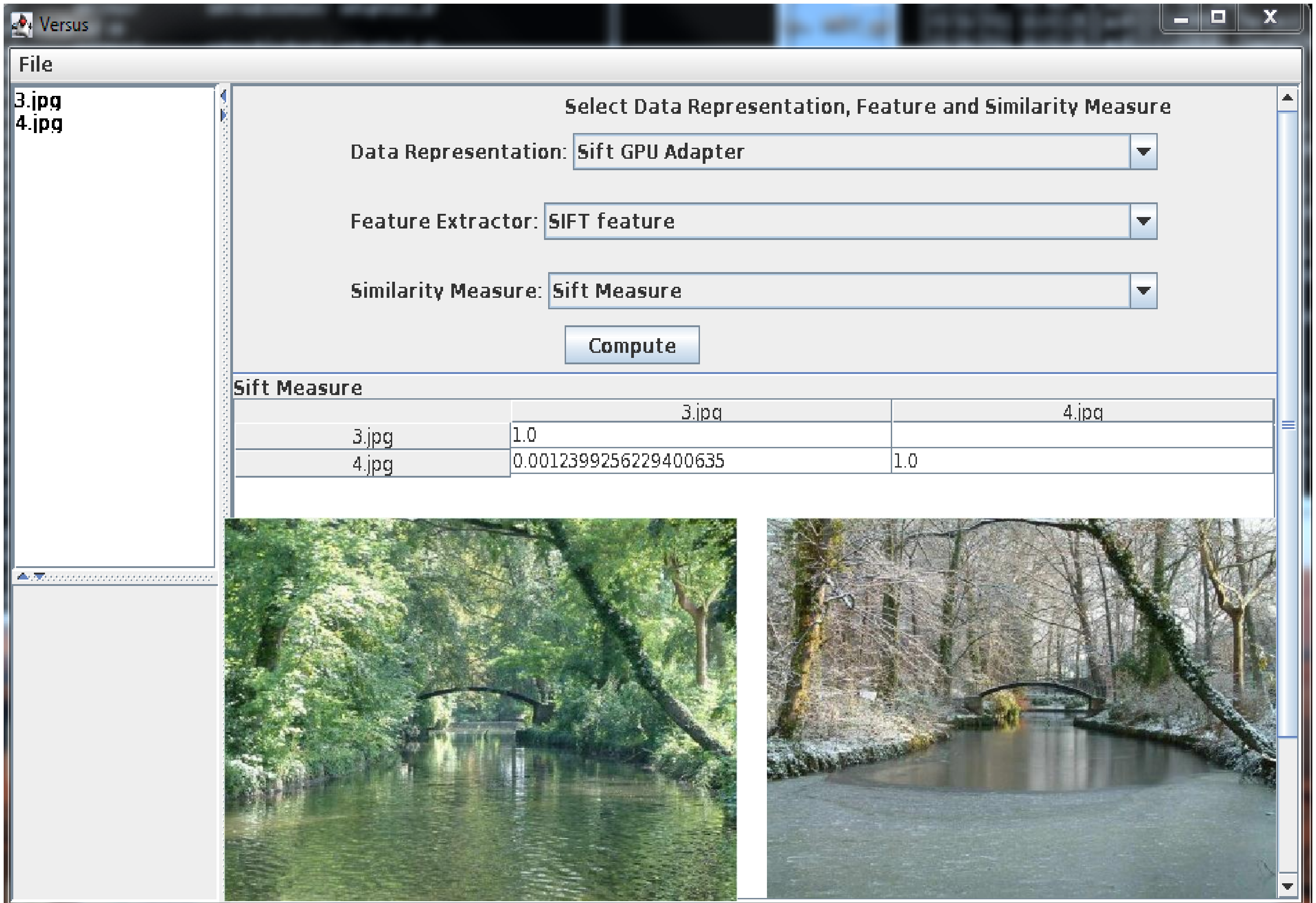
Versus example

- Two images take from the same scene but different angles
- Similarity : 0.12



Versus example

- Natural scene images from different seasons
- Similarity : 0.0012



Conclusion and Future work

- SiftGPU is good at identifying original image and its derived ones. The ability to identify the same physical scene from different images is limited.

- XtremeData database appliance evaluation

References

- [Wu'10] L. Wu, M. Storus, D. Cross, CUDA Compression Project, http://ppl.stanford.edu/cs315a/pub/Main/CS315a/CUDA_Compression_Final_Report.pdf

- [Mohan'10] Deephan Mohan, John Cavazos, Faster File Matching Using GPGPUs, SAAHPC2010

- [Bos'10]. Joppe W. Bos and Deian Stefan. *Performance analysis of the SHA-3 candidates on exotic multi-core architectures*. CHES 2010