

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	Histogram E.	Histogram F.	Pearson
BufferedImage	Texture E.	Texture F.	Subtraction
PDF Document	Array E.	Array F.	Chi Squared
ImageJ
...			

*Diagram courtesy to
Luigi Mariti*

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

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

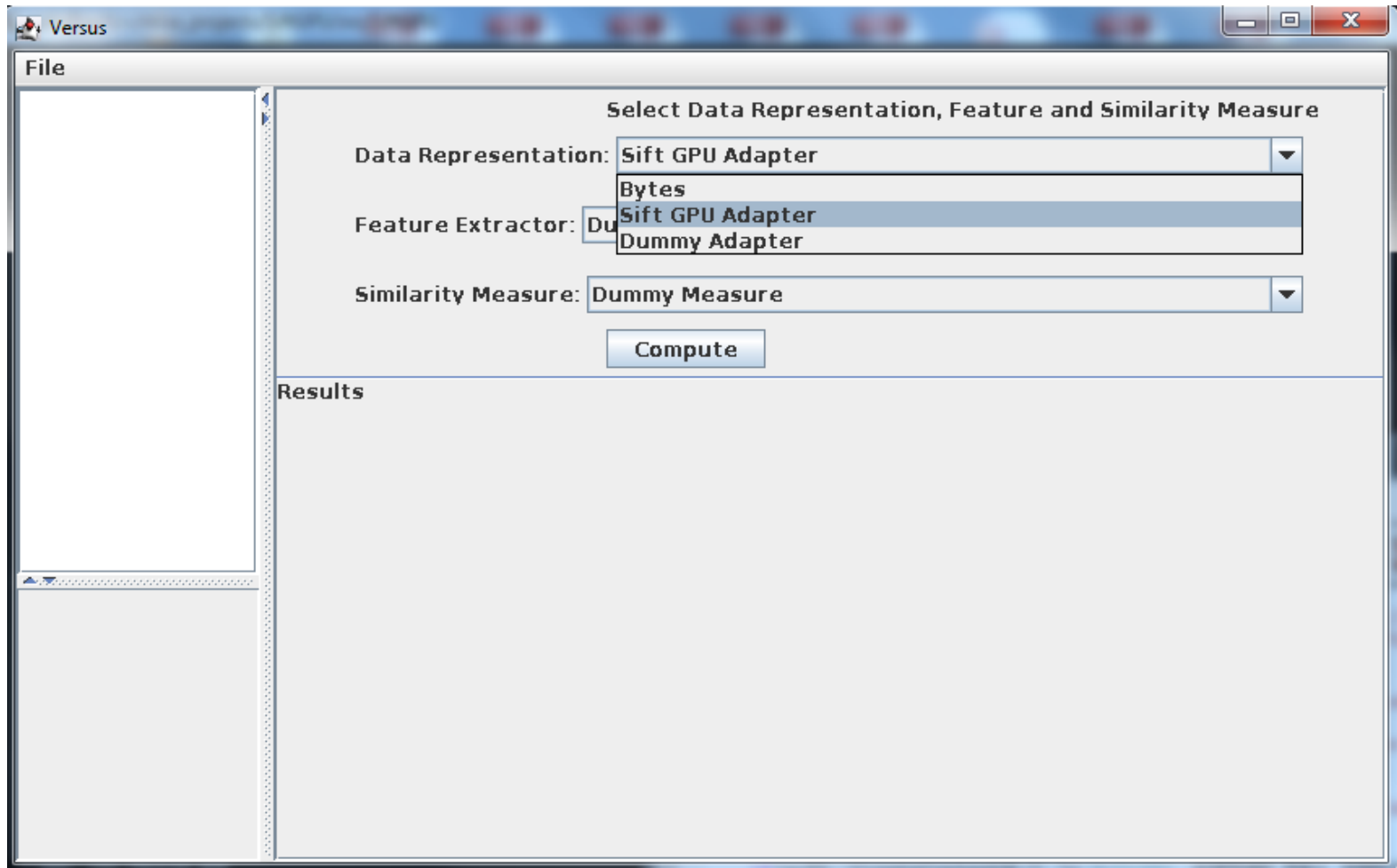
where

N1: # of features in image 1

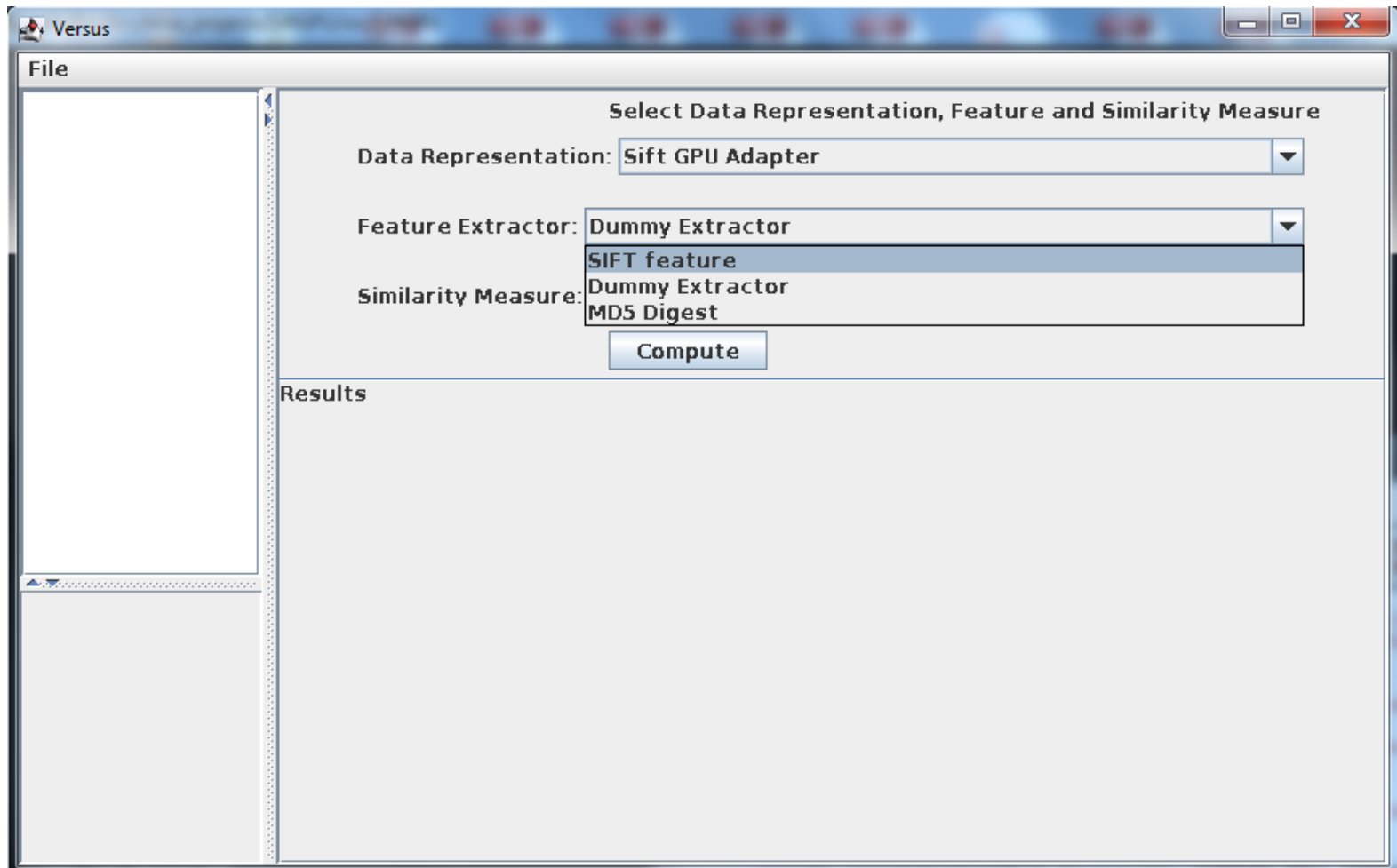
N2: # of images in image 2

n: # of matched features

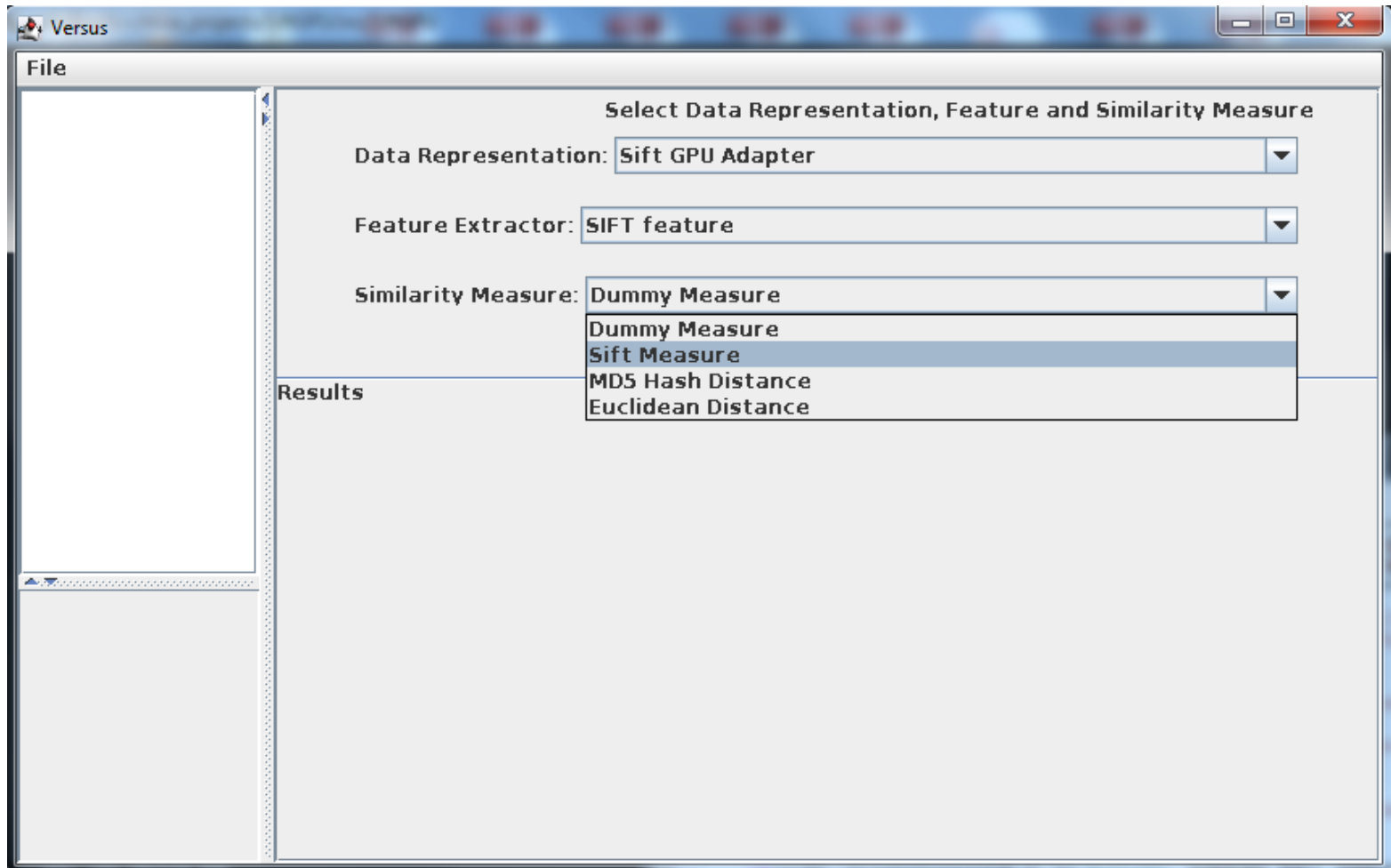
Versus example



Versus example



Versus example



Versus example

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

Versus

File

640-1.jpg
640-1-color.jpg

Select Data Representation, Feature and Similarity Measure


Data Representation: Sift GPU Adapter

Feature Extractor: SIFT feature

Similarity Measure: Sift Measure

Compute

Sift Measure		
	640-1.jpg	640-1-color.jpg
640-1.jpg	1.0	
640-1-color.jpg	0.7942212820053101	1.0



Versus example

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

Versus

File

640-1-crop.jpg
640-1.jpg

Select Data Representation, Feature and Similarity Measure

Data Representation: Sift GPU Adapter


Feature Extractor: SIFT feature

Similarity Measure: Sift Measure

Compute

Sift Measure

	640-1-crop.jpg	640-1.jpg
640-1-crop.jpg	0.9992614388465881	
640-1.jpg	0.4467625916004181	1.0



Versus example

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

Versus

File

640-1-rotate.jpg
640-1.jpg

Select Data Representation, Feature and Similarity Measure

Data Representation: Sift GPU Adapter


Feature Extractor: SIFT feature

Similarity Measure: Sift Measure

Compute

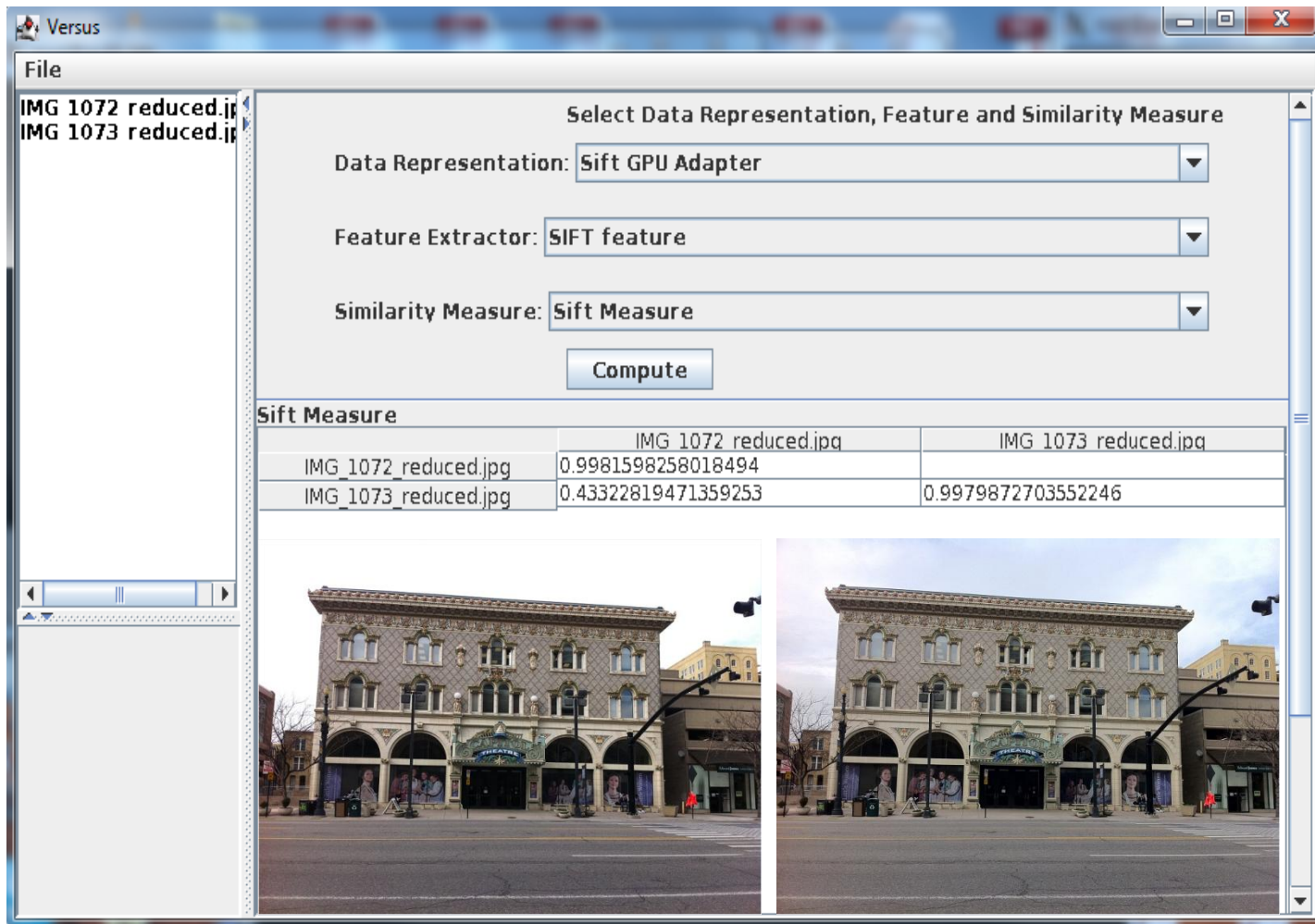
Sift Measure

	640-1-rotate.jpg	640-1.jpg
640-1-rotate.jpg	0.9996404051780701	
640-1.jpg	0.9568500518798828	1.0



Versus example

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



The screenshot shows the Versus software interface. The window title is "Versus". The "File" menu is open, showing "IMG_1072_reduced.jpg" and "IMG_1073_reduced.jpg". The main area is titled "Select Data Representation, Feature and Similarity Measure". It contains three dropdown menus: "Data Representation" set to "Sift GPU Adapter", "Feature Extractor" set to "SIFT feature", and "Similarity Measure" set to "Sift Measure". A "Compute" button is located below these menus. Below the button is a table with the following data:

Sift Measure	IMG_1072_reduced.jpg	IMG_1073_reduced.jpg
IMG_1072_reduced.jpg	0.9981598258018494	
IMG_1073_reduced.jpg	0.43322819471359253	0.9979872703552246

At the bottom of the interface, two side-by-side images of a building facade are displayed. The left image is brighter, and the right image is darker, illustrating the difference in light exposure time.

Versus example

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

The screenshot shows the Versus software interface. The 'File' pane on the left lists two images: 'IMG_1103_reduced.jpg' and 'IMG_1105_reduced.jpg'. The main window is titled 'Select Data Representation, Feature and Similarity Measure'. It contains three dropdown menus: 'Data Representation' set to 'Sift GPU Adapter', 'Feature Extractor' set to 'SIFT feature', and 'Similarity Measure' set to 'Sift Measure'. A 'Compute' button is located below these menus. Below the button is a table showing similarity results for the 'Sift Measure'.

	IMG_1103_reduced.jpg	IMG_1105_reduced.jpg
IMG_1103_reduced.jpg	0.9948281049728394	
IMG_1105_reduced.jpg	0.12468688935041428	0.9846924543380737

At the bottom of the interface, two side-by-side images are displayed. The left image shows a bookshelf with a whiteboard in the background containing handwritten code. The right image shows the same scene from a different angle, also showing the bookshelf and whiteboard.

Versus example

- Natural scene images from different seasons
- Similarity : **0.0012**

Versus

File

3.jpg
4.jpg

Select Data Representation, Feature and Similarity Measure

Data Representation: Sift GPU Adapter


Feature Extractor: SIFT feature

Similarity Measure: Sift Measure

Compute

Sift Measure

	3.jpg	4.jpg
3.jpg	1.0	
4.jpg	0.0012399256229400635	1.0



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](http://ppl.stanford.edu/cs315a/pub/Main/CS315a/CUDA%20Compression%20Final%20Report.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