Web Accessible Image Similarity Measurements

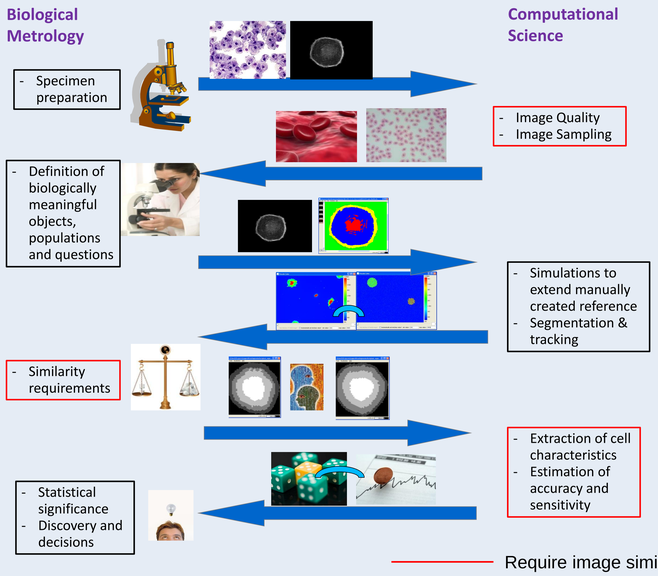
[1]Antoine Vandecreme, [1]Paul Khouri Saba, [1]Ben Long, [1]Joe Chalfoun, [1]Peter Bajcsy, [1]Mary Brady

NIST National Institute of Standards and Technology U.S. Department of Commerce

[1]Software and Systems Division
Information Technology Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899

Background

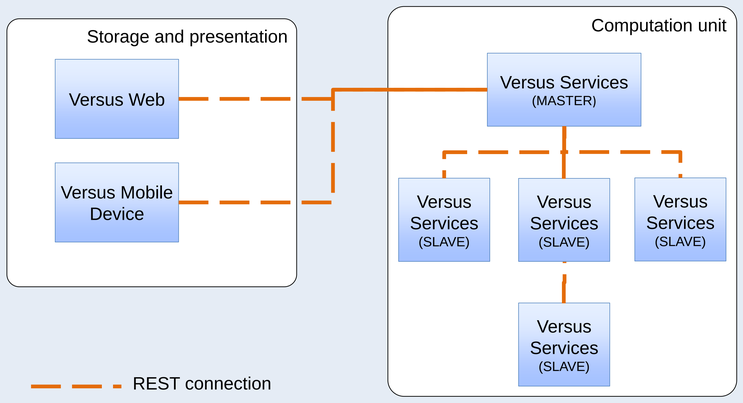
There is a need to understand the variability of biological conclusions due to the choice of a similarity metric, and due to the software quality and parameters of similarity computations. The goal of our effort is to advance high throughput and high confidence image comparisons accessed from any stationary or mobile computer device. We built a web accessible and computationally scalable system composed of image similarity metrics The similarity metrics are validated regularly by pre-configured tests. We have also added a workflow editor that allows access from a variety of platforms.



Approach

Our approach is based on organizing and evaluating image similarity metrics first according to several existing surveys of image similarities. The similarity metrics are represented by a triplet consisting of image loaders and color space representations, image descriptors, and proximity measures. The proximity measures are grouped into those that can operate on histogram descriptors, contiguous image segments, clusters of image pixels or raw pixel values. This classification of individual computations and their sub-categories allows us to build a simple tree taxonomy encapsulating image loading/representation, image characterization and comparison, and to map the taxonomy into intuitive web interfaces.

Target Architecture



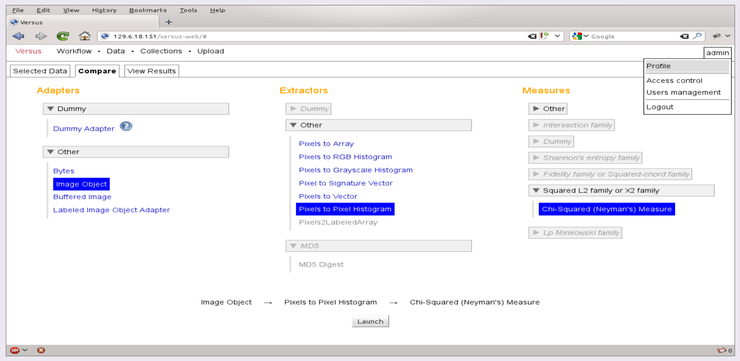
Catalog of Image Similarity Metrics

Family	Measure
Chi-Squared Family	AdditiveSymmetricChiSquared, Clark, Divergence, NeymanChiSquared, PearsonChiSquared, ProbabilisticSymmetricChiSquared, SquaredChiSquared, SquaredEuclidean
Combinations Family	AvgDifference, KumarJohnsonDifference, TanejaDifference
Fidelity Family	Bhattacharyya, Fidelity, Hellinger, Matusita, SquaredChord
Inner Product Family	Cosine, HarmonicMean, InnerProduct
Intersection Family	Czekanowski, Dice, Intersection, Jaccard, KulczynskiS, KumarHassebrookPCE, Motyka, Ruzicka, Tnimoto, WaveHedges
L1 Family	Canberra, Gower, Kulczynski, Lorentzian, Soergel, Sorensen
Lp Minkowski Family	ChebyshevLInf, CityBlockL1, EuclideanL2, Minkowski
Pixel-based Family	AdjustedRandIndex, TotalErrorRateEvaluation, TotalErrorRateTest
Shannon's Entropy Family	Jeffreys, JensenDifference, JensenShannon, KDivergence, KullbackLeibler, Topsoe

Workflow decomposition of image similarity computations

- Data selection
- Comparison method selection
- Result viewing

Web Access to Image Similarity Measurements



The Web application is based on the Google Web Toolkit (GWT) client and NCSA's Medici Multimedia Content Management System. It provides functionalities for image upload, storage and annotations. The GWT client connects to the REST API and starts computations on it. It can start comparisons on collections and save the computations results for later use.

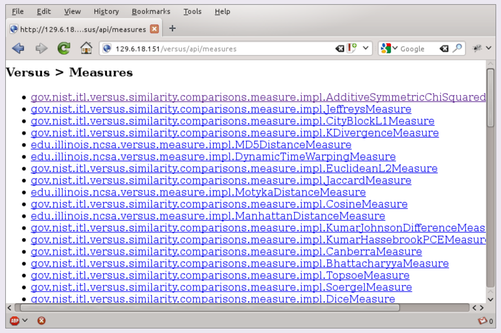
Access to Measurements via Web Services

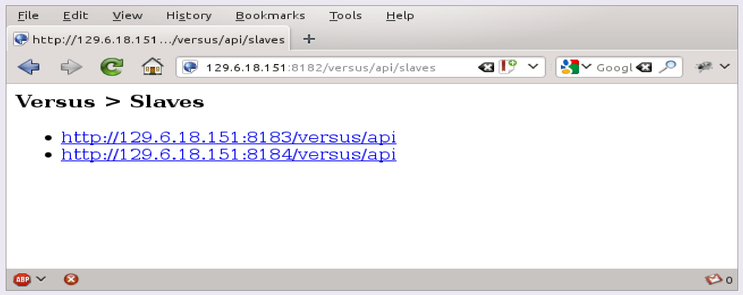
The computation system implementation is based on a RESTful API which allows to:

- Get the list of available adapters, extractors & measures.
- Submit new comparisons and get the results.
- Connect multiple slaves.

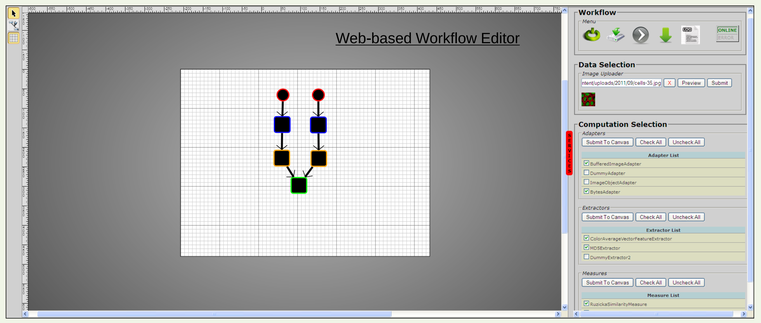
Slaves can provide:

- Computational resources
- New algorithms





Access to Measurements from Mobile Devices



The purpose of the Web-based Workflow editor is to **build** a WF with scientific data such as medical images and scientific computations (Accessed using XMLHttpRequest API), **submit** it to Taverna engine & **retrieve** the computations results. The tool is mainly written in HTML5 in order to take advantage of the cross-browser/cross-platform capabilities, and support execution on mobile devices, desktops and laptops. Other HTML5 components like SVG (Scalable Vector Graphics) and Local Storage (aka Web Storage) are also used to draw shapes, and store and retrieve objects locally.

Image Similarity Testing and Validation

Data-driven tests, one component of the overall quality testing strategy, in this case, exercise unit and failure-mode test conditions to identify errors corresponding to specified fault categories. These tests demonstrate expected failures due to image incompatibilities across different image parameters as well as cross-platform consistency of results. They detect errors triggered in 1 of 5 primary fault categories: hardware (HW), software (SW), image compatibility (Image), consistency with mathematical definitions (Math), and consistent treatment of failure conditions (Singularity). Each line in the graph depicts an error distribution for a specific test on a given platform. Together they show a cross-platform consistency of response for the implemented measures under test. Each was performed with image collections across multiple modalities (fluorescent and phase-contrast), image parameters (pixel size, dimension, type, color model) and differing platforms (Windows and Linux, with different implementations of Java).

The largest test (last 2 lines) combined all modalities and dimensions into one test. 56 synthetic images were used representing single/multi-band, multiple pixel data types (byte...double), pixel sizes (8..32), RGB and grayscale image variations. Executed as 1,166,592 comparisons across both platforms, yielding consistent results.

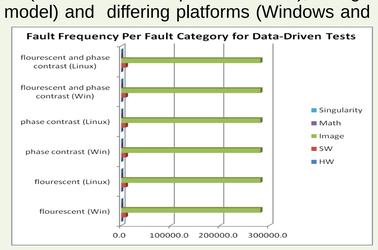
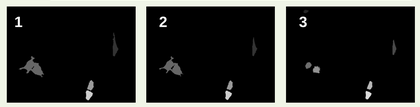


Image Similarity Results

Comparison of 3 microscope images with different similarity metrics



Adapter	Extractor	Measure	Images 1 and 2	Images 1 and 3
Buffered Image	Grayscale Histogram	Jaccard (Similarity) 1 identical, 0 different	0.9999999910825415	0.9999911129545518
Buffered Image	Grayscale Histogram	Dice (Similarity) 1 identical, 0 different	0.9999999955412707	0.9999955564575309
Buffered Image	Grayscale Histogram	Euclidean L2 (Math. Distance)	6.164414002968976	194.6021582614129

The pairwise comparisons of 750 16bit TIFF images (707KB / image) using Java desktop application (280,875 comparisons) take 50min on a Quad core Intel Xeon @ 2.80GHz with 6GB of RAM.

Execution time of pairwise comparisons of 50 images (1,250 comparisons)

Program	Execution	CPU usage
Java desktop with 1 thread	36s	36s
Java desktop with 2 threads	21s	41s
Java desktop with 3 threads	16s	46s
Java desktop with 4 threads	14s	50s
Java desktop with 5 threads	14s	51s
REST client querying server with 10 thrds	58s	18s
REST server (1 thread pool)	58s	1m11s

The usage of the REST API slows down the reading of the files due to network usage.
We plan to explore the reduction of computational times by using image caching and by managing data distribution on slave nodes.