Image Search System*

Peter Cho and Michael Yee, MIT Lincoln Laboratory

We present a prototype system which enables users to explore the global structure for digital imagery archives as well as drill-down into individual pictures. Our search engine builds upon computer vision advances made over the past decade in low-level feature matching, large data handling and object recognition. We demonstrate hierarchical clustering among images semi-cooperatively shot around MIT, automatic linking of flickr photos and aerial frames from the Grand Canyon, and video segment identification for a TV broadcast. Our software tools also incorporate visible vs infrared band selection, color content quantization and human face detection.

I. Introduction

Digital images are currently shot and stored in vast numbers. Billions of photos and video clips may now be accessed via public internet and private offline archives. But most existing imagery archives are unstructured and unorganized. Navigating through image repositories consequently requires clicking through seas of thumbnails. Aside from occasional human-tagged keywords, little connection typically exists between archived images to help users find stills or frames of interest. New search capabilities are consequently needed to mine huge electronic imagery volumes.

In this paper, we present a prototype system which enables user exploration of global structure as well as individual picture drill-down for O(10^4-10^5) images. Our system's netcentric design allows multiple analysts to cooperatively collaborate on different archive sets. Its front-end includes a web browser thin client and graph viewer thick client whose states remain synchronized. Its back-end server is based upon a database that stores imagery metadata, attributes and topological relationships. As we shall see, combined thin and thick client perusing provides a practical means for gaining comprehensive insight into large imagery collections.

Our article is organized as follows. In section 2, we first review how to generate graphs from arbitrary sets of input images. Such graphs yield clusters of similar-looking pictures that share low-level feature content. Subsequent pyramiding of graph clusters forms hierarchies for *a priori* unorganized imagery. In section 3, we describe how node groups and individual pictures may be annotated. Furthermore, image attributes such as gross camera geolocation are readily visualized via the graph viewer. Using these software tools, we search for connections between aerial and ground shots in urban and rural scenes. In section 4, we present several examples of querying digital imagery. Users can rapidly discover images collected in infrared or visible bands as well as select dominant color contents. Finally, we discuss automatic detection of continuous segments and human faces in TV video broadcasts.

II. Image Graphs

The Scale Invariant Feature Transform (SIFT) is currently one of the most popular methods for finding and labeling image features [1]. Like many other feature extractors, SIFT identifies image interest points based upon greyscale gradient content. For each interest point, SIFT generates a 128-dimensional vector whose descriptiveness significantly discriminates between image features. Standard matching techniques in computer vision enable a machine to find correspondences between SIFT features in different images [1,2]. For example, 30 SIFT matches were automatically identified for the photo pair in figure 1. The number of feature tiepoints between two input images correlates with their overall visual similarity.

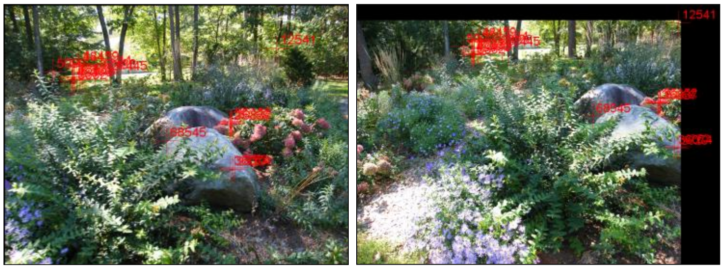


Figure 1. 30 SIFT feature matches found for this photo pair.

SIFT matching among an input set of images naturally generates an output graph. Each node in the graph corresponds to an image. Any two nodes are connected by an edge if and only if some number of their SIFT features match. Figure 2 illustrates the image graph for a set of 21 garden photos. Edge coloring in the graph is a function of SIFT match number. Hot colored edges between two image nodes indicate they share relatively large numbers of tiepoints, while cool colored edges indicate smaller numbers of SIFT links.

*This work is sponsored by the Department of the Air Force under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

978-1-4799-1365-7/13/$31.00 ©2013 IEEE

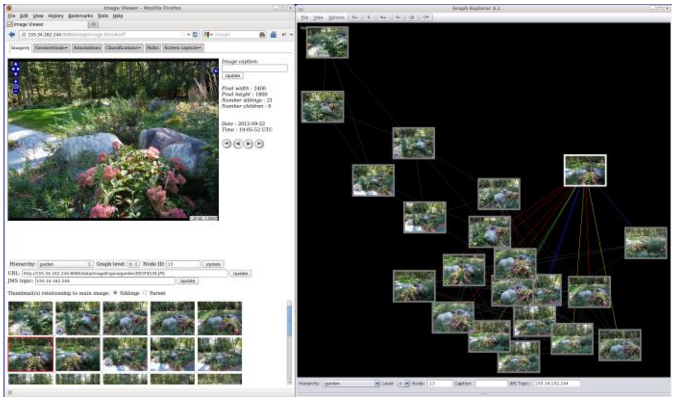


Figure 2. Search system front end displaying set of 21 garden images.

Figure 2 also portrays the front-end of our image search system. The graph viewer appearing on the figure's right displays an image collection's global structure. In contrast, the web interface shown on the left enables drill-down into individual images. When a user mouse-clicks on some thumbnail inside the browser's carousel, the corresponding node in the image graph is highlighted. Similarly, when a user clicks on a node in the graph viewer, its associated image is displayed in the browser's primary window along with accompanying metadata. The thin and thick clients thus remain synchronized as a user explores an image archive.

The front-end of the search engine is connected to its back-end via a computer network as the system diagram in figure 3 depicts. The engine's design obeys several practical software requirements. Firstly, its thin and thick clients can operate on Windows, Macintosh and Linux computers. The graph viewer is written in Java which is cross-platform, and the thin client runs in Firefox, Safari and Chrome browsers.[1] Secondly, all relationships either manually or automatically derived from imagery are saved into a Postgres database running on a single server. Stored image metadata may be retrieved at any time by any client on the computer network. The system's netcentric architecture consequently enables multi-user collaboration. Finally, no component of the system diagram in figure 3 depends upon commercial licenses. The search engine's computer codes are therefore readily deployable in a variety of settings.

Our first garden imagery set was intentionally chosen for its simplicity. The next data set we consider consists of 2328 photos shot around MIT in summer 2010 [3]. The Bundler toolkit was used to match SIFT features among all images in this second archive [4]. Image feature extraction and matching were performed on Lincoln Laboratory's LLGrid parallel computer system [5] with specially parallelized codes [6]. A total of 44K+ pairs from the 2.3K+ MIT photos were found to have 20 or more SIFT matches. After an edge list for the image graph was generated, the layout of its nodes was calculated via the Open Graph Drawing Framework [7]. The resulting image graph as well as a few representative thumbnails from the MIT photo set are displayed in figure 4.

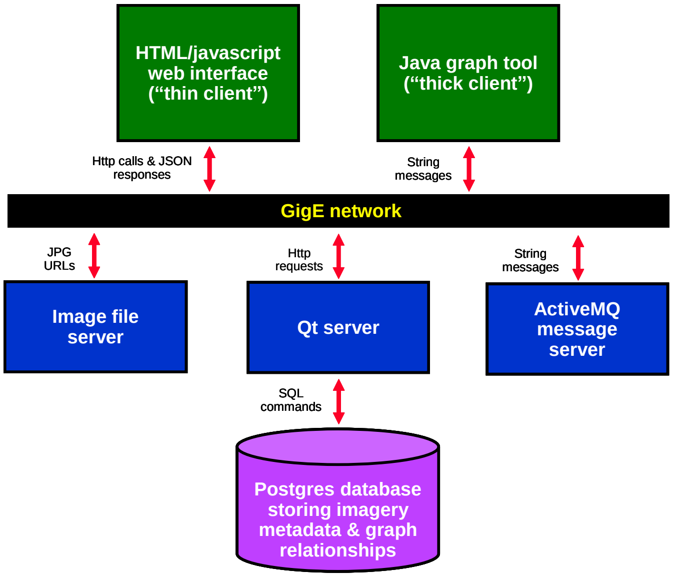

HTML/javascript web interface ("thin client")
Java graph tool ("thick client")
Http calls & JSON responses
String messages
GigE network
JPG URLs
Http requests
String messages
Image file server
Qt server
ActiveMQ message server
SQL commands
Postgres database storing imagery metadata & graph relationships

Figure 3. Image search system diagram.

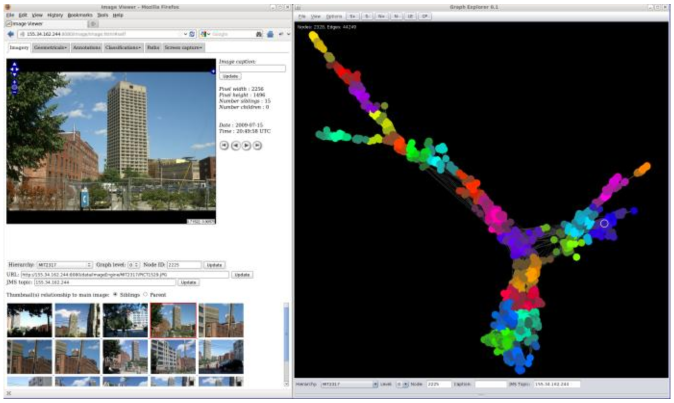


Figure 4. Node clustering and coloring for 2.3K+ photos shot around MIT in July 2010.

A high degree of overlap exists among the 2.3K+ MIT pictures. So we have employed the K-means algorithm to form 232 clusters from the 2328 input nodes. Each cluster is colored in figure 4. As there are many SIFT matches among images belonging to a commmon cluster, it is convenient to choose the node with highest degree as a cluster representative. A graph may subsequently be formed from just the representative nodes. Layout, clustering and coloring algorithms are rerun on the smaller image network. In this recursive fashion, we form a pyramid of image graphs. Nodes at level L in the pyramid all have unique parents at level L+1, and they generally have multiple children at level L-1. The hierarchical recursion is terminated when the highest-level image graph contains O(10) nodes.

Figure 5 displays the top graph in the 2.3K+ MIT photo pyramid. It is much easier to navigate its consolidated 23 members than the original graph's 2328 nodes. Moreover,

[1] No attempt has been made to support Internet Explorer.

scanning through the highest-level graph's thumbnails in either the thick or thin client viewers provides a practical means to gain an overview of the contents for an entire archive. If some thumbnail at the highest level is particularly interesting, a user may drill-down into its children, grandchildren and lower descendants inside the hierarchy. This pyramided graph scheme consequently organizes large image sets which are *a priori* unstructured.

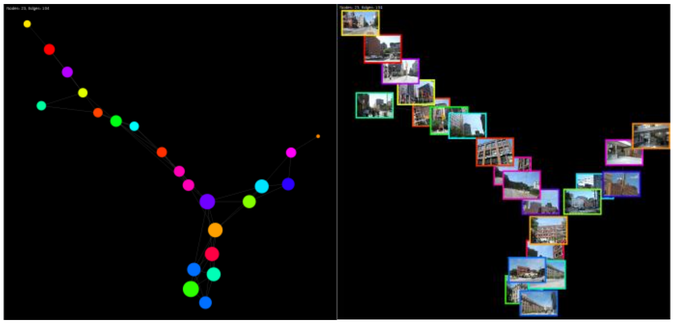


Figure 5. Highest level graph containing 23 nodes in the pyramid for 2.3K+ MIT images.

The 2328 MIT photos we have seen so far are a subset of 36243 images shot around the main campus from 2010 to 2011. If this larger set of images covered every possible viewing angle around MIT, its SIFT graph should be a single connected component. But the quasi-random nature with which photos were gathered around MIT resulted in data coverage gaps within the complex urban scene.

The graph for the 36K+ images breaks apart into multiple connected components as figure 6 illustrates. SIFT connections between the separate components are tenuous or nonexistent due to missing data. The graph viewer displays each component's ID as well as its node count. The figure thus illustrates how different parts of an image network can be annotated.

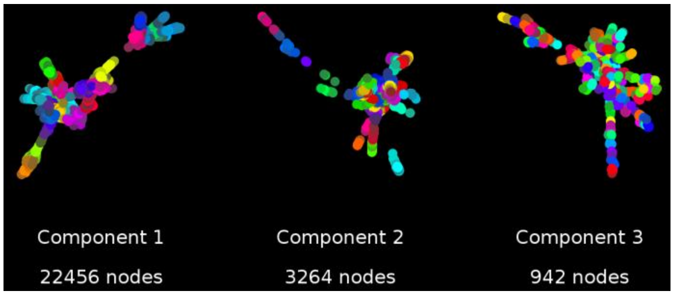


Figure 6. Largest 3 connected components of image graph constructed from 36K+ MIT images.

III. IMAGE ANNOTATIONS AND ATTRIBUTES

Our search system enables intuitive inspection of and navigation through large imagery collections. But for it to be a more practical tool, the search engine needs to enable labeling of groups of pictures as well as individual images. It should also identify photos and video frames satisfying various properties of interest. So in this section, we describe how to annotate images and view their attributes within our search system.

The thin and thick clients allow users to add captions for any archive picture. The caption is displayed alongside its corresponding node in the graph viewer. Since similarly-colored nodes look visually similar, one node's caption is generally applicable to many of its neighbors. Captioning therefore represents a simple example of knowledge propagation along the image network.

A caption encodes information about an entire picture. But it generally cannot describe multiple objects of interest nor their locations within an image plane. So the web browser interface allows users to mark points of interest in an image and persist them in the database. If another person later views a previously marked photo, the labels are automatically pulled from the database and superposed onto the image appearing in the main browser window.

It is not practical to display multiple labels for individual pictures inside the thick client. But it is useful to convey which images have been labeled with interest points. So nodes can be specially colored to indicate their counterpart images carry labels. Human analysts may then select just those pictures for inspecting or editing.

Node coloring provides a convenient means for indicating a variety of image attributes beyond user-added labels. For example, among the 36243 MIT images, 3855 are aerial video frames collected by a Star SAFIRE camera built by FLIR Systems [8]. In this case, we know *a priori* which of the 36K+ images were gathered on the ground versus in the air. The graph viewer may be instructed to recolor just the aerial nodes (see figure 7). The separate disconnected components appearing in the figure correspond to either all-ground or all-aerial images. The machine found no SIFT matches between overlapping views from significantly different perspectives. While disappointing, this result is not surprising given that SIFT feature matching generally fails for image pairs separated in out-of-plane angle by more than 45 degrees.

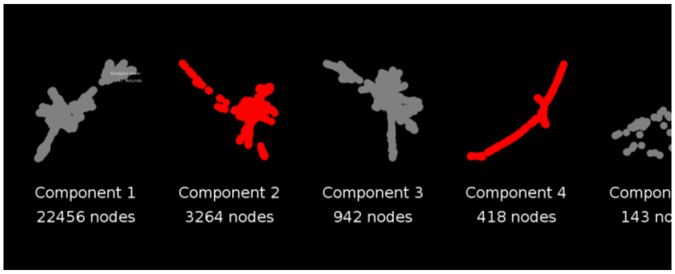
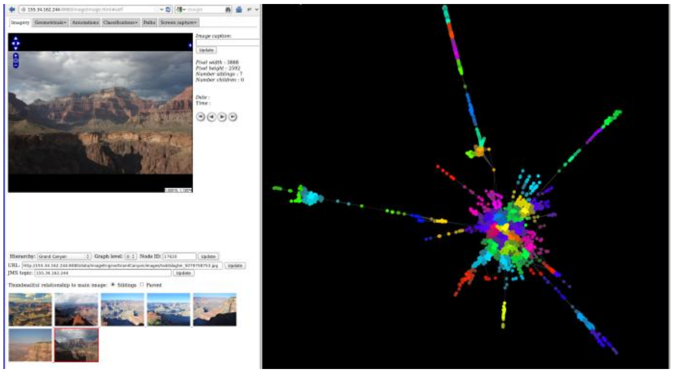


Figure 7 . Red nodes in 36K+ MIT image graph indicate aerial video frames.

It is interesting to reconsider automatic ground and aerial view matching for the qualitatively different case of the Grand Canyon. Our next set of 18479 Grand Canyon images is much more diverse than the MIT data, for it primarily consists of pictures downloaded from the flickr photo-sharing website [9]. Images harvested from the internet originate from hundreds of people shooting at different times and places. They accordingly exhibit much more variation than our MIT photos which were semi-cooperatively collected by a handful of Lincoln Laboratory volunteers over a few days. The new data set also includes 3477 aerial video frames collected by the STAR Safire camera which overflew the Grand Canyon in May 2011 (see figure 8).



Pixel width : 3888
Pixel height : 2592
Number siblings : 7
Number children : 0
Date :
Time :

Hierarchy: Grand Canyon Graph level: 0 Node ID: 17610
JMS topic: 155.34.162.244
Thumbnail(s) relationship to main image: ● Siblings ○ Parent

Figure 8. Largest connected component in Grand Canyon graph generated from flickr photos and aerial video frames.

The largest Grand Canyon connected component contains both ground and aerial views. There consequently must be a nontrivial number of SIFT matches between some subset of uncooperatively-gathered flickr images and cooperatively-collected Star SAFIRE frames. In figure 9, we observe that a tenuous link does indeed exist between a single grey-colored flickr node and several red-colored aerial video nodes.

We note that a “horseshoe” structure is visible in all aerial views directly connected to the single flickr node which acts as a bridge between aerial and ground Grand Canyon images (see figure 9a). The same “horseshoe” structure appears in the bridge photo itself (see figure 9b). We do not have camera position metadata for this or other flickr Canyon pictures. But the bridge photo looks like it was shot from the air. The “horseshoe” reemerges in the two flickr photos to which the bridge node is linked. One of the two shown in figure 9c appears to have been collected from a lower altitude than its predecessor. And the “horseshoe” structure is present within the photo of figure 9d which was clearly shot from the Grand Canyon’s rim. This last picture is densely connected to many other ground-level pictures located deep inside the largest connected component of the Grand Canyon graph.

This chain of images serves as an existence proof that it is possible to automatically connect aerial and ground views of distant targets. Because the Grand Canyon is so vast, the angular discrepancy between views of distant points from the rim and from 10,000 feet above the Canyon can be 35 degrees or less. So SIFT feature matching could just barely succeed to link imagery gathered by ground and airborne cameras.

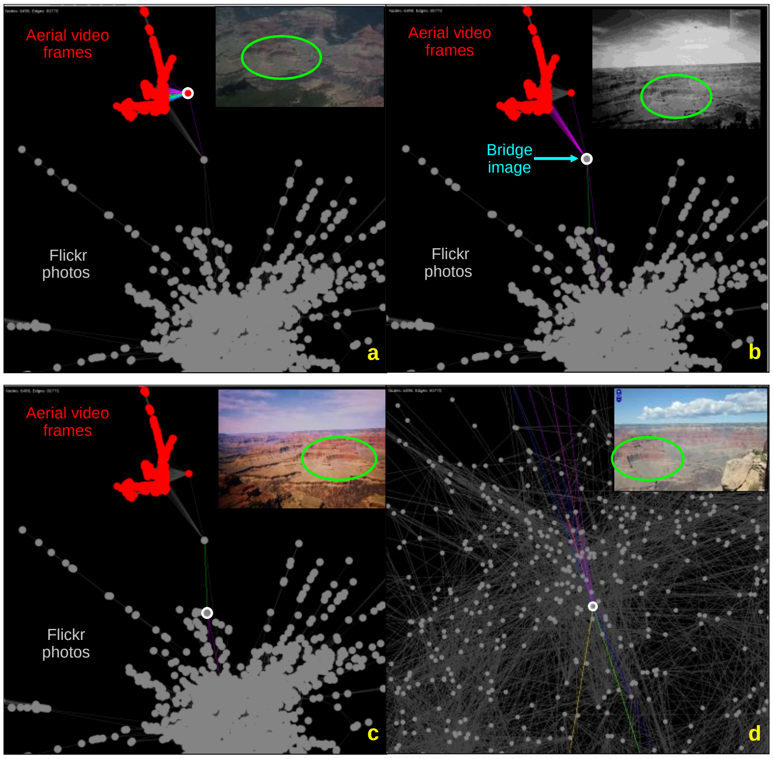


Figure 9. Link between aerial video frames and flickr photos of the Grand Canyon. Images appearing in the upper right insets correspond to nodes circled in white. Within each inset, a common “horseshoe” canyon structure is highlighted in green.

IV. IMAGE QUERIES

A. Frequency and color content

The frequency band for all of the imagery presented so far corresponds to the visible part of the electromagnetic spectrum. However, data collected at other wavelengths can also be handled by our search engine. In particular, some Grand Canyon aerial video was gathered in the mid-wave infrared (wavelength = 3-5 microns). Grand Canyon pictures accordingly have visible and infrared frequency tags in addition to aerial and ground view attributes. Using the web browser interface, one may request to see any combination of these properties highlighted in the graph viewer.

The 128-dimensional descriptors for ground target SIFT features in the infrared (IR) significantly differ from their electro-optical (EO) counterparts. So the machine found no matches between views of the same part of the Grand Canyon seen in the IR and EO bands. But there is strong SIFT overlap among the infrared images themselves. They consequently form their own connected component in the Grand Canyon image graph (see figure 10).

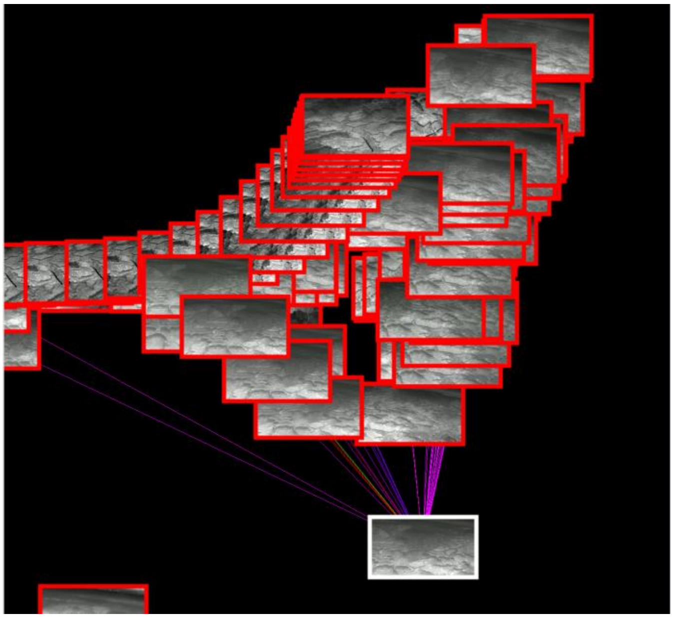


Figure 10. SIFT cluster of infrared aerial frames collected over the Grand Canyon in May 2011.

Airborne and ground view attributes for nearly all Grand Canyon images could readily be assigned based simply upon their Star SAFIRE or flickr camera origins. But JPEG stills extracted from Mpeg-4 movies do not contain any metadata indicating whether they were shot at EO or IR wavelengths. So we utilized the frames' color contents to determine their frequency classifications. Video images with nontrivial RGB spectra could only correspond to visible band pictures. On the other hand, predominantly greyscale frames were assumed to have been shot in the infrared. The only nontrivial aspect of this binary classification arises from small quantities of non-grey coloring introduced into genuine infrared pictures by video compression and JPEG formating artifacts.

The color contents of visible spectrum photos represent important features for automatic scene segmentation and object recognition (e.g. sky is usually blue, vegetation is often green, ground is normally brown, etc). Color quantization is consequently incorporated into the search system. We precalculate quantized color percentages for each archived image and store the largest values in the back-end database. Such color information provides another handle for image search. When a user requests to see all pictures whose dominant colors match specified inputs, the query results are highlighted in the graph tool. The user may further command that a quantized version of a particular image be recalculated in real time. The quantized output is then displayed in the main browser window (see figure 11).

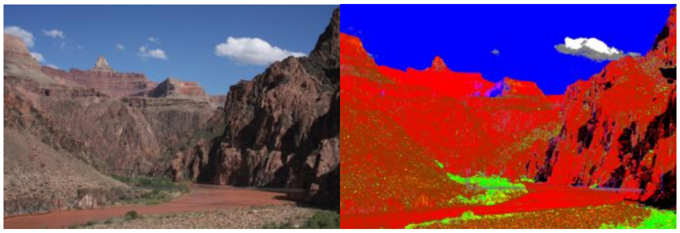
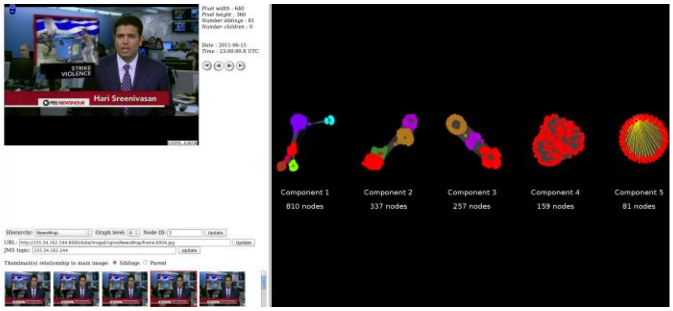


Figure 11. Color quantized version of one flickr photo from the Grand Canyon archive.

B. Video segment identification

As we have seen in the MIT and Grand Canyon data sets, our search system can analyze aerial video after it is decomposed into individual frames. It can similarly work with ground video imagery. Movie frames collected at a rapid update rate exhibit a high degree of temporal overlap by design. So SIFT matches between successive frames are generally strong. However, edited video clips often exhibit temporal discontinuities. For example, TV broadcasts include camera jumps, scene dissolves and commercial breaks. Image graphs for edited videos consequently split apart into multiple components. But within each component, edge connections between nodes are typically dense (see figure 12).



Pixel width : 640
Pixel height : 360
Number siblings : 81
Number children : 0

Date : 2011-06-15
Time : 23:00:00.8 UTC

Hierarchy: NewsWrap Graph level: 0 Node ID: 3 Update
URL: http://155.34.162.244:8080/data/ImageEngine/NewsWrap/frame-0004.jpg Update
JMS topic: 155.34.162.244 Update

Thumbnail(s) relationship to main image ● Siblings ○ Parent

Figure 12. Video frames excised at 5 Hz from a July 2011 PBS Newshour broadcast.

The frames appearing in figure 12 originate from an 8 minute section of a PBS Newshour broadcast aired in July 2011 [10]. Figure 13 zooms into the largest connected component of this clip's graph. The component exhibits several node clusters which are tightly bound by SIFT matches. The associated thumbnails displayed in the web browser carousel look like frames from a continuous movie. Separate clusters may differ from each other by relatively minor changes. For instance, one cluster primarily consists of a correspondent's head shots. Its neighbor also has frames containing the correspondent's head, but a computer graphic label appears underneath. SIFT contents for the two imagery subsets are sufficiently similar so that they reside in the same connected component. Yet they differ enough to end up in separate clusters.

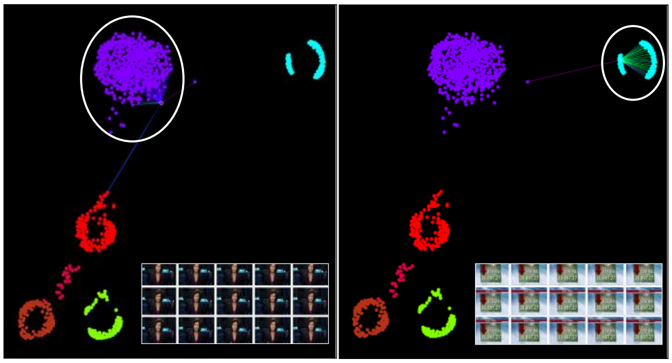


Figure 13. Clusters among the largest connected component of the PBS Newshour video clip. Lower left insets display representative frames from the clusters circled in white.

It is fun to inspect the special node in figure 13 which exhibits SIFT overlap with two highly distinct image clusters. One corresponds to stock market report frames, while the other represents a talking woman. As can be seen from the carousel insets, the images in these two clusters look totally different. So it is initially surprising that any SIFT edges were found between these two clusters. But as figure 14 reveals, the mystery video frame turns out to be a fade from one sequence to the other. The graph links therefore make sense.

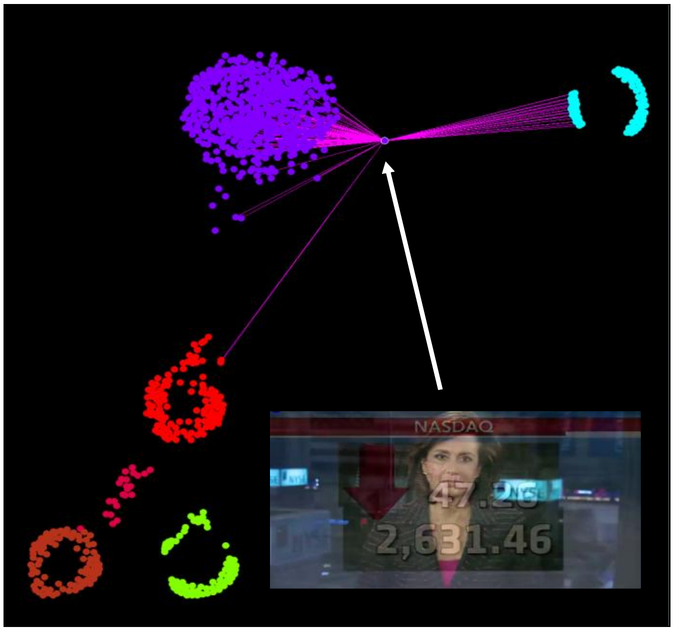
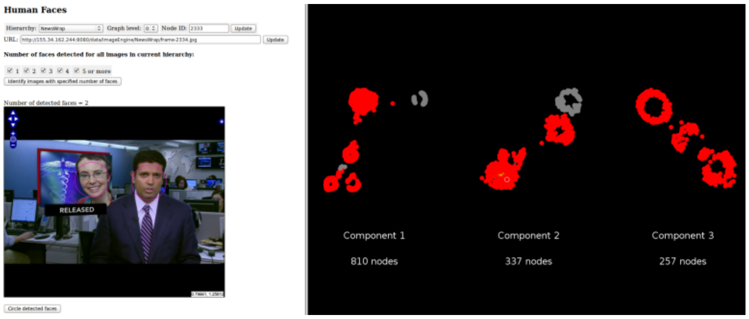


Figure 14. Node linking stock market report and woman correspondent frames corresponds to a fade between these disparate video elements.

C. Face detection

The last image search capability we consider is human face detection. Computer vision groups around the world have been working on face detection for decades, and it remains an active area of research. We have made no attempt to develop our own face detection algorithms. Instead, we have simply experimented with a few open-source face detector codes and incorporated the work of Kalal et al into the search system [11].

This particular detector requires a few dozen seconds to locate faces in each PBS Newshour frame whose pixel size equals 640x360. So we have precalculated its results for all members of this video archive. When a user issues a query, the system retrieves algorithm results from the database and colors nodes corresponding to frames with detected faces (see figure 15). It also circles detected faces in the primary browser window. Our search system enables one to specify the number of detected faces. As the number increases, this query begins to act like a crowd finder.



Human Faces

Hierarchy: NewsWrap Graph level: 0 Node ID: 2333

URL: http://155.34.162.244:8080/data/ImageEngine/NewsWrap/frame-2334.jpg

Number of faces detected for all images in current hierarchy:

☑ 1 ☑ 2 ☑ 3 ☑ 4 ☑ 5 or more

Number of detected faces = 2

Figure 15. Red nodes indicate video frames containing at least one detected human face. Image plane face locations are circled in pink within the browser window.

V. REFERENCES

[1] D.G. Lowe, “Distinctive image features from scale-invariant keypoints,” International Journal of Computer vision (2004), 91-110.

[2] R. Hartley and A. Zisserman, “Multiple view geometry in computer vision, 2nd edition,” Cambridge University Press (2003).

[3] P. Cho and N. Snavely, “3D exploitation of 2D ground-level and aerial imagery”, Applied Imagery Pattern Recognition Workshop (2011), 8.

[4] N. Snavely, “Bundler: Structure from Motion (SfM) for unordered image collections,” http://phototour.cs.washington.edu/bundler/ .

[5] N. Bliss, R. Bond, J, Kepner, H. Kim and A. Reuther, “Interactive Grid Computing at Lincoln Laboratory,” Lincoln Laboratory Journal, vol 16, no 1, (2006) 165-216.

[6] S. Agarwal, N. Snavely, I. Simon, S.M. Seitz and R. Szeliski, “Building Rome in a day,” 12th International Conference on Computer Vision (2009) 72-79.

[7] M. Chimani, C. .Gutwenger, M. Junger, K. Klein, P. Mutzel and M. Schulz, “The Open Graph Drawing Framework,” 15th International Symposium on Graph Drawing (2007).

[8] See FLIR surveillance products website at http://gs.flir.com/surveillance-products/star-safire/star-safire-III/ .

[9] See flickr photo sharing website at www.flickr.com.

[10] See PBS Newshour website at www.pbs.org/newhour/ .

[11] Z. Kalal, K. Mikolajczyk and J. Matas, “Face-TLD: Tracking-Learning Detection applied to faces,” 17th International Conference on Image Processing (2010), 3789-3792.