Visual Islands: Intuitive Browsing of Visual Search Results

Eric Zavesky
Columbia University
Dept. of Electrical Engineering
emz@ee.columbia.edu

Shih-Fu Chang
Columbia University
Dept. of Electrical Engineering
sfchang@ee.columbia.edu

Cheng-Chih Yang
Columbia University
Dept. of Computer Science
cy2184@columbia.edu

ABSTRACT

The amount of available digital multimedia has seen exponential growth in recent years. While advances have been made in the indexing and searching of images and videos, less focus has been given to aiding users in the interactive exploration of large datasets. In this paper a new framework, called *visual islands*, is proposed that reorganizes image query results from an initial search or even a general photo collection using a fast, non-global feature projection to compute 2D display coordinates. A prototype system is implemented and evaluated with three core goals: fast browsing, intuitive display, and non-linear exploration. Using the TRECVID2005[15] dataset, 10 users evaluated the goals over 24 topics. Experiments show that users experience improved comprehensibility and achieve a significant page-level precision improvement with the visual islands framework over traditional paged browsing.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Algorithms, Design, Experimentation, Performance

Keywords

Visual Islands, image browsing, visualization, image and video retrieval

1. INTRODUCTION

The importance of video indexing and search has dramatically increased as web and user-generated content continues to grow. When searching video content for a specific target, visually inspecting many lines and pages of irrelevant results actually wastes a user's time. Modern multimedia information retrieval systems return thousands of candidate results, often providing a high recall while sacrificing precision. While this approach is simple and easy to implement, the burden on the user is tremendous. This burden is exacerbated when the search interface is small with very little usable landscape, as in mobile devices. Authors in [4],[12] attempt to solve this problem by only showing the most interesting parts of an image. Using approximate human attentional models, faces, complicated textures, and non-static backgrounds are identified to be important regions of interest. After resizing and cropping out these regions, they are inherently more compact and suitable for smaller displays. This solution, although innovative, may cause some dissatisfaction if the user is instead searching for results with less object-centric content, like images of majestic scenes or a very complicated urban street.

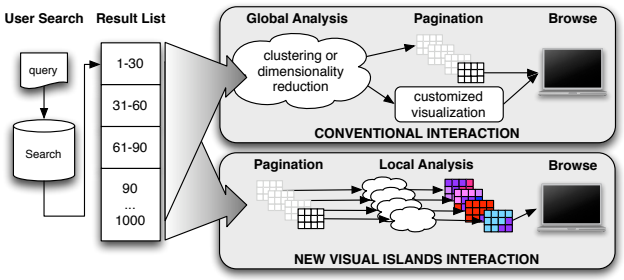


Figure 1: Comparison of conventional and new approaches using dimensionality reduction. Conventional methods analyze features for all results whereas the new algorithm minimizes disruption of original result ranking with page-based dimensionality reduction.

Works in result visualization often employ feature reduction as a technique to map images with high-dimensional raw-features to a fixed display, as demonstrated in fig. 1. To provide grounds for equal comparison in this paper, all methods are assumed to start with the results either from an executed user query or a personal. In the conventional pipeline, a system using dimensionality reduction performs a global analysis over all results and then applies a clustering or dimensionality reduction algorithm. Afterwards, either pagination or a customized display method is applied to produce user-browsable content. A second approach, proposed in this work, also performs dimensionality reduction, but only after an initial pagination of the search results. There are a few significant advantages to this alternative. First, the clustering or feature reduction algorithms are allowed to make local, dynamic choices of optimal features based exclusively on the content that is currently displayed, not all results. Second, this approach is significantly faster because there are far fewer samples (in this example 30 vs. 1000) to be analyzed and clustered. Third, instead of deriving an

CIVR'08, July 7–9, 2008, Niagara Falls, Ontario, Canada.

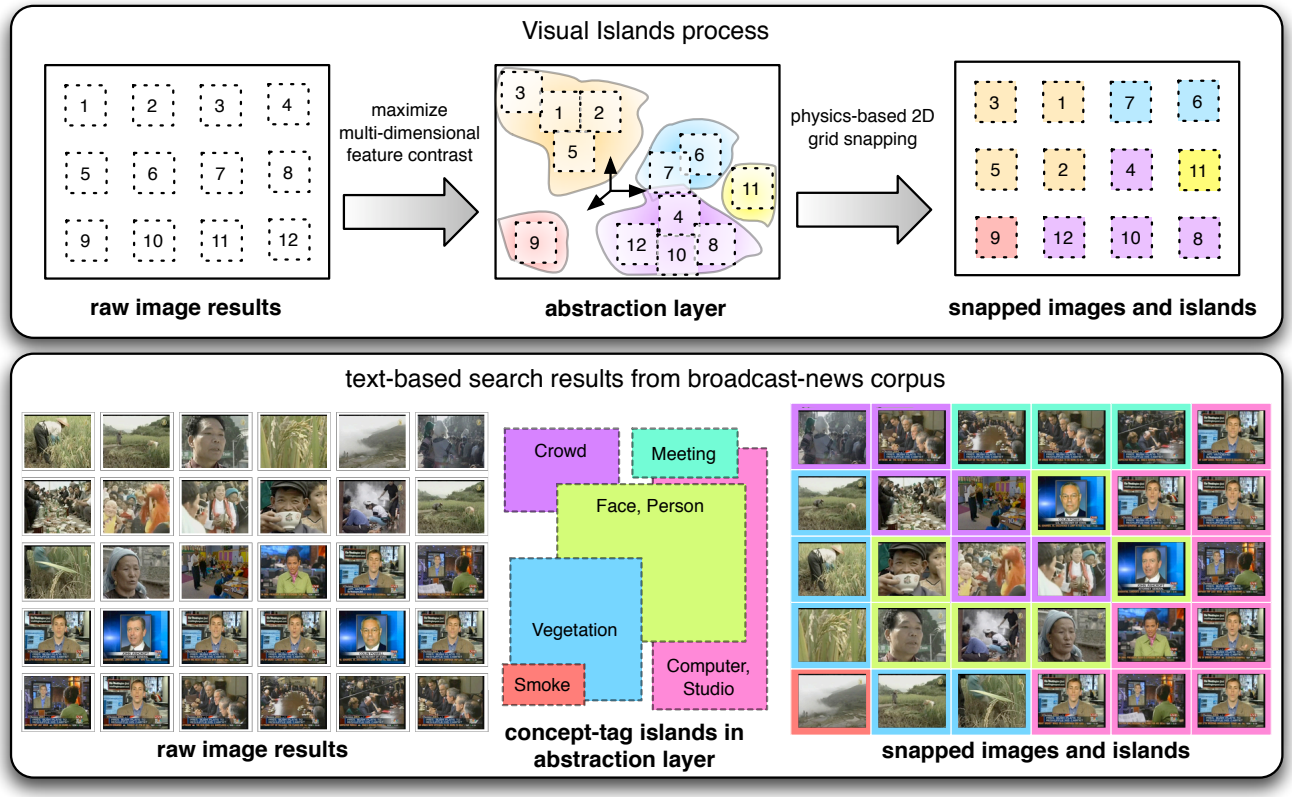


Figure 2: Visual Islands process (top) and raw and snapped results from a text-based search (bottom).

entirely new rank based on similarity in the reduced feature space, the new method roughly approximates the original rank at a coarse page level. This is advantageous because the original rank was determined by a direct query from the user instead of a ranking based on similarity in the raw-feature space alone. Additionally, analyzing results at a local page level naturally allows for non-linear result exploration by jumping between pages, which is further explored in sec. 2.3.

One less computationally tangible side effect of a challenging interactive display is the fatigue and frustration that a user might experience. In real-world situations, the time available to explore and identify relevant images for a query is often limited, so users are hard-pressed to quickly explore a large set of results. Taking this paradigm to an extreme, [7] evaluated the utility of allowing only a fraction of a second for a user to small sets of result images from a single query. Although the authors found that users were able to annotate quite a few images, the overall user exhaustion from this annotation was significant and thus not viable for most environments. Another concern for an interactive system is the layout of images, which has traditionally been in a grid format. In [17],[14] the authors explore a number of alternative display configurations to mitigate this problem. One method provides multiple search threads for the user may choose, but its main display iterates through images on a single row. While this display was shown to be useful, users may have difficulty learning how to use the interface and cope with the loss of 2D image context, as opposed to a traditional grid display, which they are more accustomed to.

In this paper, a framework and prototype implementation are presented that optimize layout configurations for a fixed-size display to improve human comprehensibility and reduce required inspection time. This method leverages similarity within an abstraction layer to more intuitively present search results to a user with only a few contrast dimensions that maximize feature contrast. This innovative framework is a significant shift from traditional paradigms that limit users to either list- or grid-based exploration of results according to their query-ranked order. In another break from convention, raw-feature analysis is performed at a page level instead of globally over all results, allowing for a fast, dynamic computation of an abstraction layer and coarsely maintaining the original ranks of the user's initial query. The rest of this paper is organized as follows. Sec. 2 gives an overview of important principle goals and how they effect modern image search applications. Sec. 3 introduces system design considerations and its implementation is defined in sec. 4. Sec. 5 and sec.6 provide experimental results and conclusions.

2. PRINCIPLES

Waste less time browsing, intuitively understand results, and allow non-linear exploration; these are the three guiding principles for this work. Although automated information retrieval systems may be improving in accuracy, there are inherent problems when directly transplanting automatically generated results into a user-search environment. The proposed system operates not only on image retrieval results from an initial query topic, but also on collections of personal images.

Fig. 2 demonstrates a few powerful traits of the visual islands framework. First, comparing images in the bottom row, one can see that the visual islands framework naturally places images with some visual (and conceptual) similarity near each other. Visual islands employs a middle layer of abstraction between raw-features and the user display to better organize images by maximizing feature contrast. Second, the tags (or concepts) chosen for the vertical and horizontal axes are usually orthogonal, indicating that the data

was strongly separable by these tags (or concepts). Finally, upon inspecting the final image, one can find several prominent visual islands. In the broadcast news example above, there are two notable islands with outdoor vegetation on the left and another with computers and a news-studio setting on the right. With these examples demonstrating how intuitive, concept-based islands can be formed, the remainder of this section is devoted to defining the principle goals of the visual islands framework.

2.1 Coherence and Diversity

Humans make relevance judgements more quickly on items that are related. However, discovering a relationship that is ideal for this judgement process is not trivial. The most simple approach is to identify images that are exact- or near-duplicates of one another. Prior work in [3],[18] uses a robust graph-based model to identify image near duplicates. These duplicates are thus be immediately grouped together so that a relevant judgement for one image could be applied to all near-duplicate images and therefore maximizing image diversity. While this method is robust for a very similar set of images, it is not suitable for a highly diverse collection for two reasons. First, the method will not identify and match the contents of two images if they share non-salient but similar backgrounds. Second, if there are no duplicates in the entire collection of images, no reliable image relationships can be established and the problem of image grouping remains unanswered. A common approach in image visualization is to arrange images by their clustered low-level features. Authors in [16] use color, texture, and structure of images and create multiple low-dimensional projections using a principal component analysis. The authors demonstrate that a deliberate spatial layout, using low-level feature projections does help to improve the identification of relevant images. Global coherence in visual features is maximized therein, but when a query topic itself is not easy to qualitatively represent using low-level features (i.e. *Find shots of a tall building (with more than 5 floors above the ground)*), this method may fail. To better handle a very large collection of diverse images, authors in [11] use a hierarchical clustering system (or self-organizing map) which ensures that both coherence (via clustering) and diversity (via a multi-level hierarchy) are maximized for a set of images. Unfortunately, even for modern search systems, the computational requirements of this algorithm are still too great. These requirements are more pronounced over a large set of results because the entire collection of images must be organized in this algorithm. In this work, high coherence and diversity are achieved with low computational cost because results are processed at the page-level (instead of globally) in small sets, which also permits the feature reduction algorithms to run quite quickly.

2.2 Intuitive Display

One problem in inspecting image results is human understanding of what the search engine was trying to judge as relevant; not even state-of-the-art query processing algorithms can directly map human query input to effective automatic searches. Instead of addressing the problem at query input, an alternative approach is adopted to assist the user in understanding image search results and more generally, sets of unordered images. For every image in a system's database, there are certain low- (color, edge, etc.), mid- (semantic concepts, contextual links), and even high-level (text annotation, tags) pieces of information that can be used to assist visualization. Methods employing dimensionality reduction, like PCA [16], for display purposes do achieve some success in that they project raw-feature similarities and differences into a 2D display. However, the usefulness of feature dimensionality reduction with PCA is diminished if focus is not given to the underlying features in the 2D space. A second approach alters the sizing and placement of images based on their similarity to a single probe image [8]. While this display may assist a user in finding images that are most similar to the probe image (like the image of a world leader) it can not help the user find or understand correlations between a many diverse images (like world leaders and their nation's flags). The proposed system analyzes and presents information about features on a page level, which aide the user in quickly understanding how the results on a page related beyond their raw-features. Two dynamically detected, intuitive dimensions that maximize abstract layer contrast are naturally aligned to a fixed-size 2D interface for display.

Surveying the larger field of multimedia processing, alternative methods were discovered that could be used for display configuration. Extending the usefulness of a PCA analysis, authors in [10] proposed that the first step in visualizing similarity within an image set should be giving images a physical representation. The authors utilize PCA for dimensionality reduction to 2D. They then add a width and height to the reduced dimensions and apply an incremental algorithm that nudges the images in different directions to eliminate any possible overlap. While this approach is a suitable complement to low-level clustering, the resulting display may not resemble the features derived from PCA due to distortions incurred from the incremental movement process, especially in regions of the 2D PCA space that are very dense with samples. Applications outside of image search, like personal photo management also provided an interesting perspective on organization alternatives. In [5], authors utilized attentional models (similar to those proposed in [8]) to crop images and fit multiple images to a single frame. Notable benefits of this photo management system like automatic placement of images on a single page break down when the result images are very diverse. This weakness stems from the process of clustering features globally instead of at a local page-scale, which dilutes the utility of clustering.

2.3 Engaged and Guided Browsing

To fully utilize a user's inspection ability, a system must be engaging and guided by user preferences. Traditional page-based navigation is time-consuming and can be boring. The process of flipping to a new page of deeper results is visually disruptive and takes time to navigate. Simultaneously, a user is more likely to become frustrated as he or she explores deeper parts of a result list because chances of finding new positive instances of the query constantly decrease. Authors in [7] create a system that automatically pushes the user through a set of results at high speeds by fixing the size and position of images so that user attention is not broken. The authors also experimented with a simple form of relevance feedback to slightly modify which images would be displayed. In [14], authors offer continuously evolving search threads in a single-row interface and users are allowed to dynamically switch threads. While both of these systems offer exciting new ways for the user to break free of linear-browsing, the interfaces themselves may be disruptive for user understanding of a query. First, the interfaces in both works are a departure from well-known grid-based displays, forcing users to battle a slight learning curve when first using either system. Second, both systems were designed to serve more as an annotation tool than as a navigation and understanding tool. This design objective is not necessarily a fault, but it places an emphasis on displaying many results very quickly instead of presenting an intuitive organization of results. Third, other than relevance judgments made within each query, there is no other way for the system to engage and guide the user through his or her search results, which adds dependence on the system, not the user, to intelligently

search a result space. An answer to this dilemma is to encourage the user to influence how the next set of results will be organized. Using a mouse click, key press, or even an eye tracker, the system is given an indication of the image that is most relevant to the user's current search path and the next set of results are modified accordingly. Guided browsing attempts to place the user at a new page that is most related to the last image marked as relevant. This approach differs from the above systems because it neither explicitly requires that the user pick a new search thread nor is the new page of results influenced by user relevance alone. Finally, although guided browsing is the same spirit as relevance feedback approaches, there is no additional computation required to handle his or her new relevance judgement because the user is only jumping to different pages of the already available result list.

3. DESIGN

With the principle goals of the proposed framework firm, different design choices to achieve these goals are given a deeper analysis in this section.

Computing the distance between a set of images is not trivial. First, there are numerous forms of raw features like meta-data (time, GPS location, user-based tags, etc.), low-level features (color, texture, etc.), and mid-level semantics, that can be used for distance measurement. To maintain a flexible and robust system design, feature input is not limited to any one type. Instead, all of these possible data types are considered as potential raw features. An *abstraction layer* is a new subspace that maps the raw feature space into a generic two-dimensional space. The mapping is constrained to a 2D representation because image results will ultimately be organized in a fixed-size display space. One important note is that there can be an infinite number of abstraction layers for a result set because they are computed dynamically for each page of results, and if the user so desires, he or she can request a new abstraction layer to explore different contrasting sets of raw features. The only requirement is that the raw-features can be numerically represented in a score vector for each image in consideration. As a proof-of-concept, experiments were conducted with both mid-level semantic concept scores as visually illustrated in fig. 2. With this representation, an appropriate distance metric and feature-reduction algorithm can be chosen, as discussed later in sec. 4.

3.1 Axis Layout

Display optimization is performed for only two dimensions of visualization, so these two dimensions should be of high utility. Referring to fig. 3a, most users are accustomed to the shown traditional grid-layout where the most relevant images are in the top-left corner and the least relevant are in the bottom-right corner. While the returned rank of images from a search engine is important, in the context of a single page, this ordering provides little useful information to the user because a user is likely to scan through all results currently displayed before navigating to a new page or altering his or her search query. Therefore, in this work, the rank order is preserved only at the coarse page level, and within each page the two dimensions of the display are utilized to display two dimensions that have maximal feature contrast within the abstraction space, shown in fig. 3b. The two contrast dimensions can be mapped back from the abstraction layer to a set of coefficients in the original raw-feature space, which can help the user understand the significance of each display direction. Additional discussion for the choice of vertical and horizontal axes is provided below.

One important choice is the assignment of the contrast axes, either vertical or horizontal. Again, observing traditional search engines, the vertical direction is the most important direction for a

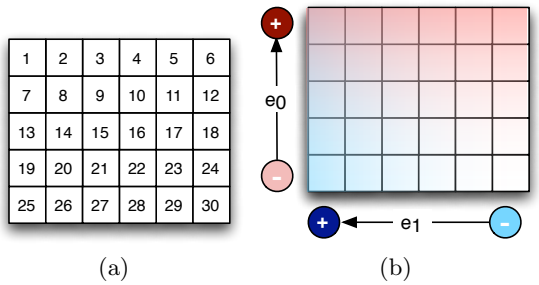


Figure 3: Visualization position for (a) traditional ranked displays and (b) a display based on two abstract layer contrast features. In (b) the highest contrast dimension is aligned vertically and the weaker dimension horizontally. The strongest intersection of both dimensions lies in the top-left corner.

user to consider, as web pages generally scroll up and down not left and right. Text results in list form are presented vertically (from most relevant to least relevant) and most image search systems have adopted the grid ordering depicted in fig. 3a. As further motivation, this observation is posited: if images are aligned to a grid, it requires less effort for a user to scan along a horizontal line because this line is parallel to the physiological plane aligned with the human eyes. When a user is forced to move from one row to the next, he or she is required to momentarily break attentional focus, so the most important images should appear at the top. Although truly based on the user's native written language, an assertion is made that users prefer to visually scan from left to right, so the most important objects should appear on the left. The consequence of these two rules is that the best (strongest) intersection of the two axes should appear in the top left and the worst (weakest) intersection in the bottom right and thus the configuration agrees with traditional ordering.

3.2 Visual Islands

A *visual island* (VI) is formed from the successful snapping of images from the abstraction layer to discrete locations in the display. Following the computation of a 2D abstraction layer, the images must then be physically snapped to a grid in the constrained 2D display space. The resulting physical space should roughly maintain the spatial layout that existed in the abstraction layer, which has been proven to be intuitive by detecting highly contrasting features, and therefore satisfies the intuitive display principle. For example, if several images share many user-tags, they should be proximal in the abstraction space. After snapping, the images should also be proximal, if not physically adjacent, and will thus constitute a visual island. More concretely, visual islands represent local clusters of images with a similar abstraction layer representation. VI's are neither restricted by size nor shape, so the prominence of a single raw feature within a visual island (i.e. the concept *sky*) or a geographic region is entirely dependent on the original image set and the context of the current page.

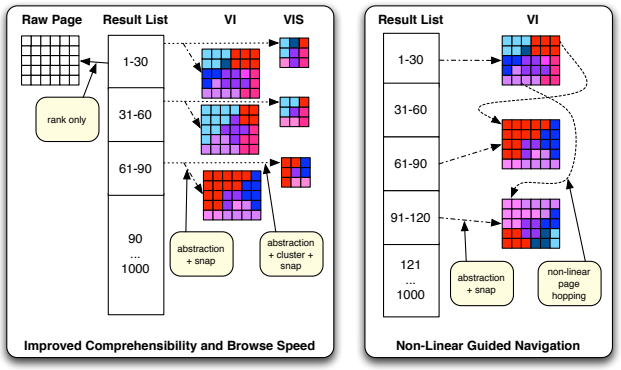
3.3 Visual Islands Summary

Visual islands summaries (VIS) represent VI's but in a smaller display size. For example, if a full-screen display occupied a 6×5 grid, then a small-screen display may occupy only a 3×3 grid, as illustrated in fig. 4a. VIS's are created in a fashion similar to a normal VI. The one exception is that before a new abstraction layer is computed, all images for a page are clustered according to their raw-feature values. Various clustering methods may be used, but in

the current implementation the k-means clustering algorithm was chosen, where k is always set to the total number of cells in the VIS to be generated. The very small initial sample count (only the total number of results on a single page) guarantees that the clustering process will run quickly, which is achieved with only a single k-means iteration although the results of this iteration may not be perfect. An alternative to clustering is performing a constant sub-sampling from a full VI computed for the same page. However, the simplicity of sub-sampling may decrease image diversity (by sampling multiple times from the same visual island) and it generally weakens the intuitiveness of the original VI's contrast features because there are images available to show a gradual transition along a contrast dimension.

3.4 Non-Linear Navigation

To fulfill the principle goal of guided user browsing, a method to dynamically rearrange the result images is provided. In an interactive environment, one can use the VI or VIS display to dynamically reorganize displayed pages with a bias towards the user's selection; this process is called *island hopping*. Unlike traditional relevance feedback. We limit the reorganization to changes in the order of displaying individual pages. The specific question is which remaining page should be shown given the relevance feedback provided by the user in the currently displayed page. This non-linear navigation is instantly available, because no additional relevance labels are evaluated in a new query. The page hopping criterion should be designed so that the subsequent page allows the user to explore deeper into the dimension that he or she is interested in. Details of the hopping algorithm will be discussed in section 4. Later, each list of pages is ordered according to the projected location of each image, again with the top-left corner receiving the highest preference, as illustrated in 4b. This method still allows the user to exhaust the entire set of query results as he or she chooses different images to follow, but the order in which the results is presented is entirely based on the image path of the user. This navigation technique allows the user to break free from traditional page-by-page linear browsing and jump to a set of results more related to the user's preferred result.



(a) (b)

Figure 4: Examples of (a) Visual Island and Visual Island Summary formation and (b) Non-Linear Navigation over pages.

4. IMPLEMENTATION

The visual islands system employs three core representations: raw features, a 2D abstraction layer, and a display space. As each representation is generic, several different data types at different levels (low-level features vs. mid-level concepts vs. semantic tags) can be used at will. Mindful of the design discussions in sec. 3, the following sections document the exact algorithms used in the current implementation of visual islands.

4.1 Abstraction Layer

The term *abstraction layer* is defined as the projection of any raw-feature modalities into a 2D coordinate representation. Many methods have been discussed to perform this projection: low-level clustering, principal component analysis [16], and even hierarchical clustering [11]. With these available tools, the system evaluates two similar techniques to discover two contrast feature vectors (hereafter referred to as e_0 and e_1) in the abstract layer: entropy based detection and singular value decomposition (SVD). In both of the following discussions, the resulting geometrical representation of the abstract layer's top-left corner is $\begin{bmatrix} e_1 & e_0 \end{bmatrix}$, as motivated by sec. 3.1.

$$
\begin{aligned}
H_i(X) &= -\sum_{x\in 0,1} p(x_i = x) log_2[p(x_i = x)] \\
e_0 &= \underset{i}{\operatorname{argmax}}\, H_i(X) \\
H_{i,j}(Y|X) &= -\sum_{x\in 0,1}\sum_{y\in 0,1} p(y_j = y, x_i = x) \\
&\qquad * log_2[p(y_j = y|x_i = x)] \\
e_1 &= \underset{j}{\operatorname{argmax}}\, H_{e_0,j}(Y|X)
\end{aligned}
$$

Figure 5: Entropy-based method for choosing two maximally contrasting dimensions over distributions of binary features.

Entropy-based detection is a simple process that analyzes value distributions of the raw-features and selects one dimension that produces the highest entropy (denoted as e_0). The rationale of choosing the highest entropy is to find an informative dimension in which the samples exhibit a great variability, instead of concentrating in a narrow range. For all computations in this implementation, a simple binary quantization threshold is applied to raw-feature values into only two values, presence or absence, although this setting could be modified in future system revisions. The probability of seeing value a in dimension i of a feature vector x is defined as $p(x_i = a)$. Following fig. 5, the point-wise entropy for dimension i is computed over all samples X and all possible values, $a \in 0, 1$. In the current implementation the point-wise entropy and conditional point-wise entropy are used to select the two contrasting dimensions e_0 and e_1. First, using a point-wise entropy computation, e_0 is found. To determine the second contrast dimension, e_1, an additional computation is performed to find the highest conditional entropy, given all possible values and probabilities of e_0.

An SVD factorization provides a set of matrices that quantify relationships between raw-features and image samples. It is a very fast, reproducible algorithm used here to compute a low-dimensional projection of the high-dimensional raw-features. As matrix-based linear algebra algorithm, a basis matrix is computable that can be used to map low-dimensional features back into the distributions in the raw-feature space. These distributions can help the user understand the newly computed contrast dimensions and their display space layout. The first prominent use of SVD for dimensionality reduction was used in a latent semantic indexing (LSI) technique for text document indexing [6]. Given a matrix of n samples and d raw-features, SVD factorizes the data into three component matri-

ces:

$$A_{(d\times n)} = U_{(d\times r)}S_{(r\times r)}V^T_{(n\times r)}, \quad (1)$$

where U represents raw-feature to projected feature affinity, S represents affinity among projected features, and V represents the document to projection affinity. First, to achieve a reduced dimensionality, zero out all diagonals of the S matrix starting from the index of the desired dimension; i.e. if $r = 5$ but the system only requires two dimensions, then $S_{i,i} = 0, i \geq 2$ using a zero-based index. Second, to project the raw-features into the lower dimensional space, A^*, compute $A^* = A^T US$ and truncate to keep only the first two columns. Finally, to retrieve the distributions over for the raw-features represented by each contrast-projected dimension, use the new SVD basis for each direction: $e_0 = \begin{bmatrix} 0 & 1 \end{bmatrix}(SU^T)^T$ and $e_1 = \begin{bmatrix} 1 & 0 \end{bmatrix}(SU^T)^T$.

4.2 Island Snapping

Island snapping is process that converts the unconstrained 2D coordinate system of the abstraction layer into a fixed, grid-cell display configuration while preserving the spatial configuration provided by the abstraction layer as closely as possible. The most direct way to snap images located in the 2D abstraction space to a grid is a brute-force quantization, where each image is assigned to the grid cell that it is closest to. For example, if the coordinates from the abstraction layer are taken as the center of an image, i, only a simple Euclidian calculation is needed to discover the closest grid cell, c.

$$dist(P_i, P_c) = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} \quad (2)$$

$$c = \underset{c\in C}{\operatorname{argmin}}\, dist(P_i, P_c) \quad (3)$$

However, a direct use of this approach will often distort the abstraction layer's spatial layout and can create different snapped results based on the order in which the images are evaluated. Thus, a more disciplined algorithm, hereafter referred to as the *corner snapping* algorithm, is proposed that snaps images by iterating over cells, which have a fixed position and can thereby be ordered systematically.

Revisiting the principle goals in sec. 2, one observes that the final island snapping stage must result in an intuitive layout of image results. First, all potential cells in the visual island grid C are ranked by their decreasing Euclidean distance to the center of the display; the regular geometry of the grid will introduce many cells that are equidistant, but this problem is addressed later. As discussed above, the system must determine the order of the cells in the grid to process. This order is critical because in each iteration the closest image to the processed cell will be snapped. Alternative ordering, such as zig-zag ordering from top-left to bottom-right, were also explored, but the visual results were far less satisfying. Second, a state variable for each image indicating whether it has been snapped or not, $lock$, is initialized to false. Now, the algorithm proceeds to iterate through all of the ordered cells, identifying the unlocked image i that is nearest to that cell, snapping the image to its new location, and setting the image's lock state repeating until all images are locked. Although not guaranteed, the corner snapping algorithm usually preserves the relative relations in the spatial layout available in the abstraction layer quite accurately. The algorithm is deterministic and will exactly reproduce the same results in every run.

Early experimentation demonstrated that the frequency of very close or overlapping images was quite high when dealing with abstraction layers derived from high-dimensional binary tag vectors. The sparsity inherent in binary tag vectors created near-duplicate raw-features that greatly diminished the visual effectiveness of the abstraction layer and snapping algorithm. Although computationally unavoidable in the abstraction layer, improvements could be made to the corner snapping algorithm that better preserved the context of the images in the abstraction layer, instead of relying only on an image's 2D placement. After studying physical and contextual models, a link between all images, similar to an elastic string was envisioned and implemented in the algorithm, as illustrated in fig. 6. Suppose that image i is to be snapped into grid cell c. If a non-elastic string between i and j is unbreakable, then j will move along a computable vector $\vec{v}$ towards cell c. This position can be found with eq. 4, where $D_{ij} = dist(P_i, P_j)$ between the original i and j positions, as defined in eq. 2.

$$P^*_j = P^*_i - \frac{|D_{ij}|}{\vec{v}}(P^*_i - P_j) \quad (4)$$

With the use of a string model, the spatial layout of the images is better maintained (and emphasized) as the images are snapped to different cells.

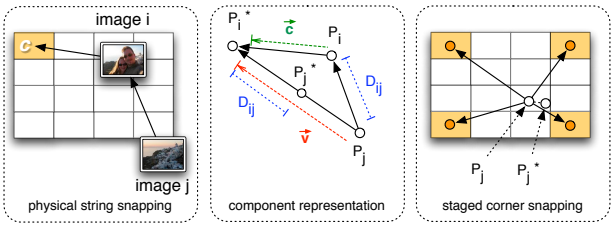


Figure 6: Physical example and component representations of iterative corner and string snapping. (a) If image i is attached to image j, (b) then j will move along a computable path between the old position of j (P_j) and the new position of i (P^*_i). (c) Corner snapping applies movement from multiple images simultaneously so that P^*_j is not biased by any one point in the next snapping iteration.

Fortunately, the original corner snapping algorithm is easily adapted to one that accounts for string relationships between images; the algorithm below provides a pseudo-code listing for the refined algorithm. There are two important revisions to the corner snapping algorithm regarding snapping in order to incorporate the addition of strings. First, instead of applying a string pulling after every corner snap, the string-based updates are staged until all cell positions with equal distance from the display center are completed, as shown in fig. 6c. This intentional staging process guarantees that unlocked images will not haphazardly jump from corner to corner and instead are pulled simultaneously from multiple snapped points, emulating the stretching out of a net or sheet of fabric at all corners. Second, to reduce the likelihood of the zero-sum movement for an image, an exponential resistance, Ω, is applied to the string snapping step; as resistance increases (between zero and one), the influence of a distant corner snap is exponentially dampened. A fortuitous side-effect of this extra resistance is that groups of images that almost perfectly overlap in the abstraction layer are effectively pulled apart with each round of string snapping.

4.3 Island Hopping

Visual island formation is an innovative step towards a more intuitive display of image results. However, the system can be augmented by a mechanism that facilitates dynamic exploration

Algorithm 1 $string\_snap\_image(rows, cols, I, \Omega)$

$C \leftarrow order\_cells\_by\_distance(rows, cols)$
for all $i \in I$ **do**
  $lock_i = false$
end for
$D_c = -1$
$corners \leftarrow \emptyset$
for all $P_c \in C$ **do**
  $P_i = find\_closest\_unlocked(P_c, I, lock)$
  $lock_i = true$
  $corners \leftarrow i$
  $P_i^* = P_c$
  if $(\Omega < 1)$ & $(D_c > 0)$ & $(dist(C_i, P_i) \neq D_c)$ **then**
    for all $lock_j \neq true$ **do**
      $P_j^* = 0$
      for all $i \in corners$ **do**
        $\vec{v} = dist(P_i^*, P_j)$
        $D_{ij} = dist(P_i, P_j)$
        $P_j^* = P_j^* + P_i^* - (1 - exp(-\frac{D_{ij}}{|\vec{v}|} * \Omega))(P_i^* - P_j)$
      end for
      $P_j^* = \frac{P_j^*}{count(corners)}$
    end for
    $corners \leftarrow \emptyset$
  end if
  $D_c = dist(C_i, P_i)$
end for

of results driven by user interest. After the user indicates one or more images as relevant (by clicking, for example) these images can be used as probes to guide subsequent navigation, which satisfies the final principle goal of guided browsing. Two different approaches were explored during implementation, but only the second approach elegantly complements other parts of the visual islands framework.

A straight-forward technique uses two similar permutations: rank the similarity between the probe image and a low-level centroid computed for each page or rank the average similarity between the probe image and every image on a page. Although numerically different, both permutations of this technique place too large of an emphasis on the low-level similarity of an image. The set of results within a page may retain their order, but now the user is limited to searching for images that look the same. Another technique best leverages the abstract layer by ranking pages based on the placement of the probe image in each page's abstract coordinates, illustrated in fig. 7. Applying previous conclusions from sec. 3.1, when a page places an image in the top-left of the display space, it has strong contrast features for both dimensions in the abstraction layer and that page potentially has high relevance to the probe image. The contrast features in the abstraction layer of each page $(\{e_0^{(1)}, e_1^{(1)}\}, \{e_0^{(2)}, e_1^{(2)}\}, \cdots, \{e_0^{(N)}, e_1^{(N)}\})$ are computed independently, so it is not possible to directly compare the projected 2D positions of the image. However, as fig. 7 illustrates, when all other images for a page are projected into this space, a computation can be made that analyzes the normalized distance between the probe image (after projection) and the top-left corner. Sorting by ascending distance, the best order of pages for the probe image can be found. Thus, the page jumping technique embraces both a CBIR spirit (low-level features are projected into an abstract layer) and the sprit of abstract layers in this work (normalized distance from the projected 2D space), which combine to accurately detect the page with the most similar visual island.

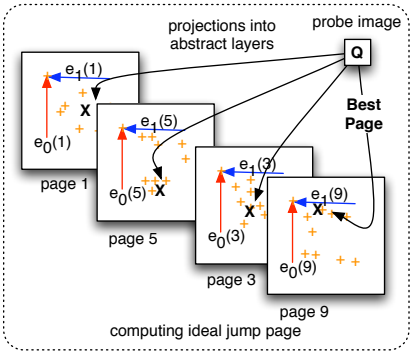


Figure 7: Example of best jump page computation utilizing abstract layer projection for each page $(\{e_0^{(1)}, e_1^{(1)}\}, \{e_0^{(3)}, e_1^{(3)}\}, \{e_0^{(5)}, e_1^{(5)}\}, \{e_0^{(9)}, e_1^{(9)}\})$, which are all dynamically computed and different.

5. EVALUATION AND EXPERIMENT

To conduct experiments, the entire TRECVID2005 dataset was used in several ways. This dataset was chosen because it has both a large collection of ground-truth human-annotated labels (*development* data with over sixty-one thousand annotated images) and because it provides ground truth for 24 query topics defined by NIST. The TRECVID[15] yearly evaluation has spurred innovation and opportunity because it focuses on several core tasks of image and video processing. Although the TRECVID2005 dataset may be considered slightly homogeneous (broadcast news recorded in three languages), it is sufficient for experiments in this work, which focus on validating a new approach to user interfaces. Two tasks were created to frame the experiments in this work: one task evaluates intuitiveness and comprehensibility and another that measures user speed and precision when presented with different interface layouts. The different layouts considered were a baseline layout (the grid layout of an ASR search over story segments), a corner-snapped layout and a string-snapped layout (see sec. 4.2) A total of 10 users with varied experience in multimedia search and retrieval were randomly paired with one of 24 query topics from the TRECVID2005 topics. Across all experiments, the same user can only be exposed to a query topic once, which prevents prior experience with the topic from affecting subsequent user decisions. Users were asked to participate in at least three experiments for each of the two tasks, but some users completed more. All experiments were conducted using a web browser of the user's choice, HTML and javascript interaction, and image layout results that are precomputed before all experiments.

5.1 Speed & Accuracy

To answer the question of coherence and diversity, the system was evaluated for the speed and accuracy of user annotation. The evaluation's assertion is that after intuitively organizing search results, users will be able to discover positive images for a specific query topic more accurately and more quickly.

To measure the speed and accuracy of the system, pages of results from a story-based query are generated using the baseline (default) order, the corner snapping algorithm, and the string snapping algorithm. The user is asked to annotate one topic at a time for 15 pages under each of the three layout configurations. The user is sequentially shown all pages using for a single layout and allowed to break for a short amount of time. Using a simple activity logger, the system tracks how much time it takes the user to navigate through each set of pages and which images were annotated as relevant. The system provides statistics independently, but they are

aggregated first by user and layout below for a more general study of this experiment. To allow for an accurate annotation, unbiased by user familiarity with a topic, the system allows users to skip a topic if he or she feels that the topic is too vague or unfamiliar.

	baseline	corner-snap	string-snap
Speed Analysis (time between labels in seconds)			
$t_{e \in E}$	3.86 ± 2.20	3.99 ± 2.35	3.67 ± 1.84
$t_{e=0}$	5.30 ± 2.37	4.46 ± 3.20	4.40 ± 0.88
Average Performance Analysis (measured at each page)			
precision	0.0883	0.0900	0.0963
recall	0.0064	0.0065	0.0069

Table 1: Mean precision (measured at the depth of each page) and mean elapsed time t between positive labels on only the first ($e = 0$) or over all image views ($e \in E$) measuring improved speed and accuracy from visual islands.

Table 1 describes the results for this experiment. A total of 18 unique query topics were evaluated by 8 unique users. While there were a total of 24 topics available, the lower number of completed topics is acceptable because users were permitted to skip topics they are unfamiliar with, which is a requisite for this experiment.

For speed analysis, the action log was scanned to compute the time elapsed between each positive annotation, initialized to zero for each new page of results; a log entry was created when each page of results is loaded and a log entry is created for each annotation. The times reported for all image views, $t_{e \in E}$, indicate that use of either snapping algorithms decreases the time required by users to find relevant images. An image viewing occurs when the user sees a page of results from any of the layouts (baseline, corner-snap, or string-snap). Users may artificially reduce annotation time by remembering previous views of results in a single topic. To discount this effect, the mean elapsed annotation time is also computed for the first viewing of results in a topic ($e = 0$) and is included in table 1. While these times are averaged over fewer users, they demonstrate a clear benefit from the corner- and string-snapping algorithms. The reduced time required for identifying relevant results, as shown by all time measurements, can be attributed to a layout that is easier for users to quickly inspect exclusively due to organization by related mid-level semantics (concepts); low-level image features were not processed for this experiment because only manual concept labels were used.

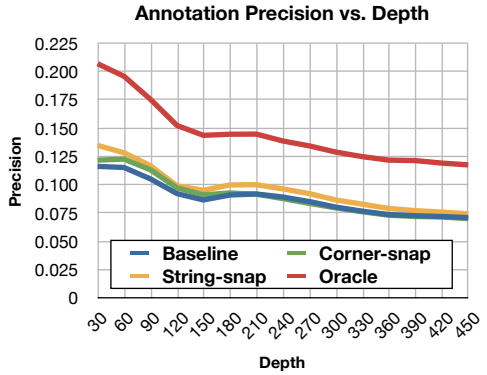


Figure 8: Annotation precision at many depths. Oracle plot indicates perfect (ground-truth) annotation.

To analyze performance benefits from the proposed layouts, user precision and recall (at each page depth) is also computed. Measurements are made at page depths instead of at each image because the layout of results using the visual islands algorithm are no longer defined in a traditional row-scan method. It should be noted that the relatively low values for precision and recall are not based on user performance, but instead the results of the initial story-based query; the performance of all methods (and the ground-truth itself) are plotted in fig. 8 below.

Fig. 8 demonstrates the benefits of the string-snapping algorithm in terms of improved precision. While not perfect (the oracle line), both proposed methods demonstrate some improvement over the baseline. An interesting trend is that as the depth of analysis increases, the benefits from either algorithm seem to decrease. This trend is explained by the observation that the quality (relevance) of results and the similarity between individual image results (which is best leveraged in the abstract layer) from the original list both decrease at higher depths.

5.2 User Evaluation

The question of intuitiveness is only truly answerable by system users themselves. While this is a subjective question, a rating system can be employed to capture user opinions contrasting the different available display configurations. Here, an assertion is made that users prefer a more intelligently organized result display over a baseline, or unstructured display. Specifically, the experiment solicits subjective opinions for intuitiveness and comprehensibility and performs an objective measurement of how well the user understood the results. For both parts of the experiment, TRECVID2005 development data (not test data) was used because it contains a complete set of manual labels for 374 concepts, which allows the abstraction layer to achieve the most accurate projection of raw features (sec. 4).

In the first part of this experiment, users are presented with five randomly chosen result pages from a single query topic, one at a time. For each set page of results, all three displays are vertically included on a single HTML result page. The vertical order of the display configurations is randomly chosen for each page so that there is no bias toward a single display. While the result sets are bound to a single query topic, the topic text is not revealed to the user because this experiment was created to judge the intuitiveness and comprehensibility of the algorithm itself, results regardless of the specific query. The users are then asked the two questions below and are allowed to input a score of 1-5, where 1 is the best. Values entered for these questions may be repeated (i.e. more than one system can be rated 5 or 1).

Q1 How well does this organization help you comprehend this set of images?

Q2 How intuitive is this organization of images (i.e. are image groupings useful and/or related)?

Additionally, to discern a preference for the different display configurations themselves, users are also asked to rank the different displays from best to worst. No time limitation is enforced for this activity.

Results from this task are documented in table 2. Unfortunately, among 10 unique users, no strong opinion is clear from the users' responses. For both questions and the final rating, users generally preferred the baseline approach. While a subsequent discussion is included in sec. 5.2.1, the conclusion of the subjective questions in this test are either not statistically significant or show a minor preference for the baseline layout approach.

	baseline	corner-snap	string-snap
Subjective Survey			
Q1	1.354 ± 0.291	1.750 ± 0.247	1.752 ± 0.261
Q2	1.434 ± 0.253	1.652 ± 0.301	1.780 ± 0.273
rank	1.721 ± 0.336	2.009 ± 0.143	2.037 ± 0.124
Memory Game (average user performance)			
precision	0.915	0.931	0.869
recall	0.625	0.595	0.642

Table 2: Subjective and objective (memory) evaluation results comparing the intuitiveness, comprehensibility, and preference for methods using visual islands.

In the second part of this experiment, a memory game is created to measure how well users can observe and retain knowledge about a set of results. First, a random page and random layout configuration is chosen by the system; the query topic providing these results is the same as the topic evaluated in the subjective part. Once all of the image results have finished loading, the user is given 30 seconds to inspect the corresponding layout after which the images are hidden and the memory game is shown. In the memory game, a checkbox is created for each of the concepts under consideration (16 for this experiment). The user must check/enable the concepts that he or she remembers from the last page of shown results.

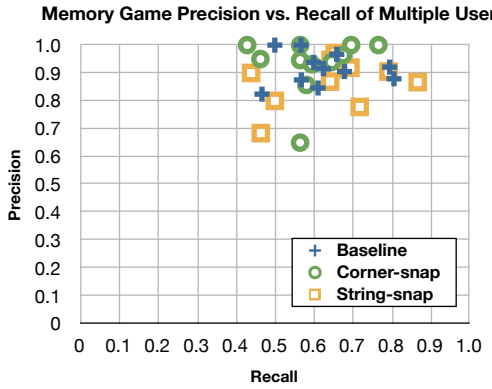


Figure 9: Precision and recall of all users of the memory game. Each point represents a user's average performance for the specified layout.

As seen in both table 2 and fig. 9, the corner-snapping algorithm was shown to generally improve precision in the memory game. Half of all users scored a precision of 0.9 or higher in all experiments, indicating that users were generally correct when they indicated a concept was present. Recall scores, which quantify how well the user identified all of the present concepts. On average, users scored higher on recall with the string-snap algorithm, which means that the layout of the string-snapping algorithm does generally increase the comprehensibility of a set of results. Informal, post-experiment interviews reveal that some users did not understand the definitions of certain concepts so they never considered them present in the memory quiz, which may explain why many users had very low individual recall scores.

5.2.1 *Battling the Baseline Bias*

Upon deeper analysis of the actual image results of these experiments, a unique property of the baseline results surfaces. In all baseline queries, a text-based search over transcripts from automatic speech recognition was used. The smallest atomic search unit, called a *document* in the text-search community, is usually a single recognized *phrase* of speech when using ASR. However, using the framework developed in [9],[3] the text documents in these experiments are actually textual *stories*, with an average duration around three minutes. The direct effect of this larger document size is that images that belong to the same story will have the same relevance score in the text-based search. Consequently, image results that may not have been scored similarly otherwise, are ranked and positioned near each other in a visual rendering. Thus, a nearness due to temporal proximity is displayed in the baseline results, whereas the corner- or string-snapped methods may break the temporal relationship in favor of more intuitive visual islands within a page of results. This behavior may artificially increase the user's preference for the baseline layout instead of the latter methods.

6. CONCLUSIONS AND FUTURE WORK

In this work, a novel organization algorithm for interactive result display called *visual islands* was proposed and evaluated. Founded by a few principle goals, this algorithm successfully offers the user an alternative method of displaying results that is more coherent, intuitive, and engaging through guided browsing. This increased comprehensibility of image results is achieved by organizing images in a way that maximizes their informativeness (e.g. entropy or latent semantics) in one of two abstract contrast dimensions. The display methods were examined for improved speed and accuracy in user browsing and improved precision based on a dynamic page jumping algorithm. Finally, and perhaps most importantly, subjective evaluations consistently preferred the proposed display method over the traditional, unorganized displays.

As a proof-of-concept, search results over a broadcast news corpus were evaluated in the experiments and one example of these results is shown in fig. 2. Although mid-level semantic concept scores and user provided tags were used to form visual islands in this work, the algorithm is general enough to accommodate just about any raw-feature and still offer a more amenable display configuration. This novel, low-cost algorithm will be employed in future interactive display experiments as a easy way to more intuitively present image search results.

Although many options were evaluated informally during the implementation of the visual islands algorithm, some additional avenues of exploration are left for future work. One question about which method to choose (high entropy detection vs. SVD) for abstraction layer computation remains open. There are clearly instances where raw-feature data is too sparse to be useful in SVD formulations, so the choice can be automated based on statistics in the data. Along these lines, additional projection methods that preserve raw-feature similarity in small local clusters can be evaluated. One popular projection method employing local optimization is locality preserving projections (LPP), as described in [1]. Finally, there is a large use potential for visual islands in continuously scrolling displays to replace page-based displays. Using the disciplined approaches in this paper, the visual islands algorithm can be iteratively evaluated to develop a continuously evolving display based on images that the user has seen and the current relative position in a result list.

7. REFERENCES

[1] Deng Cai, Xiaofei He, Jiawei Han, " Regularized Locality Preserving Projections with Two-Dimensional Discretized Laplacian Smoothing". Department of Computer Science Technical Report No. 2748, University of Illinois at Urbana-Champaign (UIUCDCS-R-2006-2748), July 2006.

[2] C.C. Chang and C.J. Lin, "LIBSVM: a library for support vector machines". *http://www.csie.ntu.edu.tw/ cjlin/libsvm*, 2001.

[3] S.F. Chang, *et al.*, "Columbia University TRECVID-2005 Video Search and High-Level Feature Extraction". In *NIST TRECVID workshop*, Gaithersburg, MD, 2005.

[4] Li-Qun Chen, Xing Xie, Xin Fan, Wei-Ying Ma, Hong-Jiang Zhang, He-Qin Zhou, "A visual attention model for adapting images on small displays." Microsoft Research Technical Report # MSR-TR-2002-125, 2002.

[5] Jun-Cheng Chen, Wei-Ta Chu, Jin-Hau Kuo, Chung-Yi Weng, Ja-Ling Wu, "Audiovisual Slideshow: Present Your Journey by Photos." In *Proceedings of the 14th annual ACM international Conference on Multimedia*, 2006.

[6] S. T. Dumais, G. W. Furnas, T. K. Landauer, and S. Deerwester, "Using latent semantic analysis to improve information retrieval." In *Proceedings of CHI'88: Conference on Human Factors in Computing*, New York: ACM, 281-285, 1988.

[7] Alexander G. Hauptmann, Wei-Hao Lin, Rong Yan, Jun Yang, Robert V. Baron, Ming-Yu Chen, Sean Gilroy, and Michael D. Gordon, "Exploring the Synergy of Humans and Machines in Extreme Video Retrieval". In *International Conference on Image and Video Retrieval*, 2006.

[8] Hao Liu, Xing Xie, Xiaoou Tang, Zhi-Wei Li, Wei-Ying Ma, "Effective Browsing of Web Image Search Results". In *Proceedings of the 6th ACM SIGMM International Workshop on Multimedia Information Retrieval*, 2004.

[9] Winston Hsu, Shih-Fu Chang, "Visual Cue Cluster Construction via Information Bottleneck Principle and Kernel Density Estimation". In *International Conference on Content-Based Image and Video Retrieval*, Singapore, 2005.

[10] Xiaodi Huang, Wei Lai, "Force-Transfer: A New Approach to Removing Overlapping Nodes in Graph Layout." In *25th Australian Computer Science Conference*, 2003.

[11] J. Laaksonen, M. Koskela and E. Oja , "PicSOM - self-organizing image retrieval with MPEG-7 content descriptors." In *IEEE Transactions on Neural Networks: Special Issue on Intelligent Multimedia Processing 13*, No. 4, pages 841-853, 2002.

[12] Yu-Fei Ma, Xian-Sheng Hua, Lie Lu, and Hong-Jiang Zhang, "A Generic Framework of User Attention Model and Its Application in Video Summarization." In *IEEE Transactions on Multimedia Journal*, 2005.

[13] G. Nguyen and M. Worring. "Interactive access to large image collections using similarity-based visualization". In *Journal of Visual Languages and Computing*, 2006.290(5500):2323âĂŞ2326, 2000.

[14] Ork de Rooij, Cees G. M. Snoek, and Marcel Worring. "Mediamill: Semantic video browsing using the RotorBrowser". In *Proceedings of the ACM International Conference on Image and Video Retrieval*, Amsterdam, The Netherlands, July 2007.

[15] A. F. Smeaton, P. Over, and W. Kraaij, "Evaluation campaigns and TRECVid". In *Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*. Santa Barbara, California, USA, October 26 - 27, 2006.

[16] Qi Tian, Baback Moghaddam, Thomas S. Huang, "Visualization, estimation and user modeling for interactive browsing of image libraries". In *International Conference on Image and Video Retrieval*, 2002.

[17] Marcel Worring, Cees G. M. Snoek, Ork de Rooij, Giang P. Nguyen, and Arnold W. M. Smeulders, "The MediaMill semantic video search engine". In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Honolulu, Hawaii, April 2007.

[18] Dong-Qing Zhang, Shih-Fu Chang. "Detecting Image Near-Duplicate by Stochastic Attributed Relational Graph Matching with Learning". In *ACM Multimedia*, New York, New York, October 2004.