Display Optimization For Image Browsing

Qi Tian[1], Baback Moghaddam[2], Thomas S. Huang[1]

[1]Beckman Institute, University of Illinois, Urbana-Champaign IL, USA 61801
{qitian, huang}@ifp.uiuc.edu
[2]Mitsubishi Electric Research Laboratories, Cambridge MA, USA 02139
baback@merl.com

Abstract. [*]In this paper, we propose a technique to visualize a multitude of images on a 2-D screen based on their visual features (color, texture, structure, etc.). The resulting layout will automatically display the mutual similarities of the viewed images. Furthermore, audio features, semantic features, or any combination of the above can be used in such a visualization. The original high dimensional feature space is projected on the 2-D screen based on Principle Component Analysis (PCA). PCA has the desired property of being simple, fast and unique (i.e. repeatable) and the only linear transformation that achieves maximum distance preservation in projecting to lower dimensions. Furthermore, we have developed a novel technique for solving the problem of overlapping (obscured) images shown in the proposed 2-D display. Given the original PCA-based visualization, a constrained nonlinear optimization strategy is used to adjust the position and size of the images in order to minimize overlap (maximize visibility) while maintaining fidelity to the original positions which are indicative of mutual similarities. A significantly improved visualization of large image sets is achieved when the proposed technique is applied.

1 Introduction

Traditional browsing and navigating in a large image database is often disorienting unless the user can form a mental picture of the entire database. Content-based visualization can provide an efficient approach for image browsing and navigation for large image databases.

In this paper, we proposed a content-based visualization technique for image browsing and navigation. The visual features used for content-based visualization are discussed in Section 2 and the proposed visualization technique is presented in Section 3.

In Section 4, a novel optimization technique for solving the overlapping problem is introduced. Example visualizations, showing maximally visible layouts, are shown in Section 5.

2 Visual Features

There are three visual features used in our system: color moments, wavelet-based texture, and the water-filling structural features.

Color is one of the most widely used visual features for content-based image analysis. It is relatively robust to background complication and independent of image size and orientation [1]. To represent color, we choose the HSV color space due to its de-correlated and uniform coordinates. Color histograms are the most commonly used color

[*] Appear in the proceedings of *2nd International Workshop on Multimedia Databases and Image Communications (MDIC'01)*, Sep. 17-18, 2001, Amalfi, Italy. The figures in this paper are best viewed in color and at higher resolution at http://www.ifp.uiuc.edu/~qitian/mdic01/

feature representation. While histograms are useful because they are relatively insensitive to position and orientation changes, they do not capture spatial relationship of color regions and thus, they have limited discriminanting power. Stricker and Orengo [2] showed that characterizing a 1-D color distribution with the first three moments is more robust and more efficient than working with color histograms. The mathematical foundation of this representation is that if we interpret the color distribution of an image as a probability distribution, then the color distribution can be uniquely characterized by its moments. Therefore, a 9-dimensional color feature vector (3 moments for each color channel, HSV) is extracted and stored from every image in the database.

Texture refers to the visual pattern with properties of homogeneity that do not result from the presence of a single color or intensity [3]. It contains important information about the structural arrangement of surfaces and their relationship to the surrounding environment. To represent texture, we used the wavelet texture representation proposed by Smith and Chang in [4]. Specifically, we feed an image into a wavelet filter bank and decompose the image into three wavelet levels, thus having 10 subbands. Each subband captures the feature of some scale and orientation of the original image. For each subband, the standard deviation of the wavelet coefficient is extracted. Therefore, a 10-dimensional texture feature vector is extracted for each image in the database.

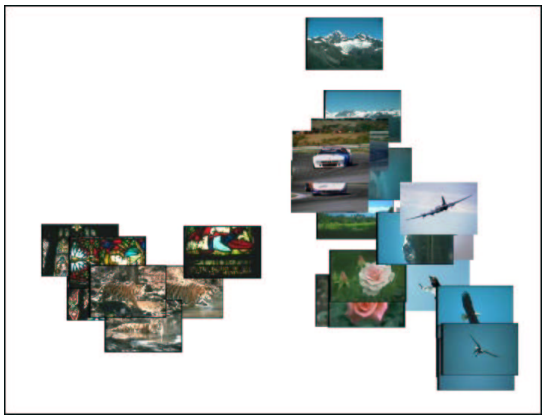
Structure is a feature between texture and shape. It is more general than texture or shape in that it requires neither uniform texture region nor a closed shape contour. In this paper, we used so-called *water-filling* structural feature [5]. For water-filling structural feature, we use a suitable edge detector and extract eighteen (18) elements from the edge maps, including *max fill time, max fork count, etc.* For a complete description of this structural feature vector, interested readers are referred to [5].

3 Content-Based Visualization

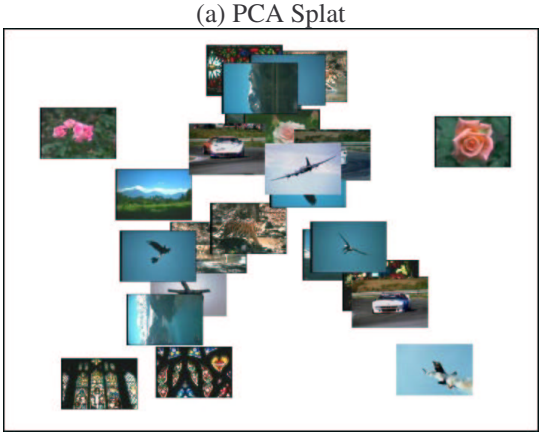
One of our central focuses is augmenting a user's perception in order to grasp a large information space that cannot be perceived by traditional image browsing methods, e.g., images are either randomly displayed or tile-based displayed. In this section, we propose a technique that visualizes the mutual similarities of a set of images on the 2-D screen based on visual features. We believe having an idea of the "context" of the current query can suggest where to go next in image browsing and navigation because the user can form a mental picture of the entire database [6].

In our experiments, the 37 visual features (9 color moments, 10 wavelet moments and 18 water-filling features) are pre-extracted from the image databases and stored off-line. The 37 visual features are formed into a single feature vector and projected to the 2D space based on Principle Component Analysis (PCA) [7]. PCA is a very fast linear transformation that achieves the maximum distance preservation when projecting from high to low dimensional feature spaces. The PCA layout is denoted as a "PCA Splat". The mutual distance between image locations respects their visual similarity, thus forming visual groupings of image clusters.

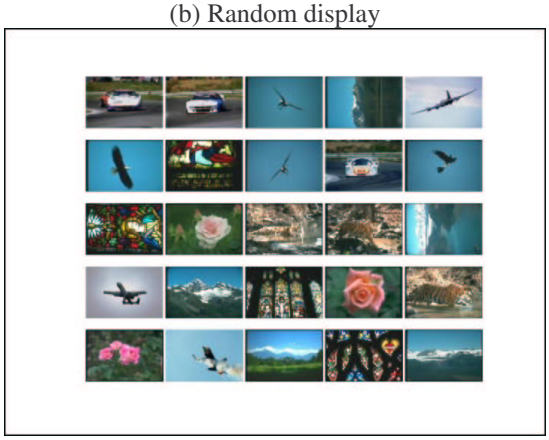
Figure 1 (a) shows an example of PCA Splat for 25 images from a natural image database. Traditional displays, e.g., random display and tile-based display are shown Figure 1(b) and (c), respectively. Clearly similar images are close to each other in a PCA Splat. PCA Splat also conveys information about all $\binom{n}{2}$ similarities between images while traditional displays (like tile-based rank-ordered browsers) cannot provide such 2^{nd}-order information.



(a) PCA Splat



(b) Random display



(c) Tile-based display

Figure 1. Different display schemes

We can also project images onto a 3-D space for more advanced visualization. The features used can be individual visual features, audio features, semantic features (keyword annotations) or any combination of the above, which we are currently experimenting with.

Multidimensional Scaling (MDS) is an alternative approach to perform dimensionality reduction for visualization. MDS [8] is a nonlinear transformation that minimizes the stress between high dimensional feature space and low dimensional space. However, MDS is rotation invariant, non-repeatable, and is about 3 order slower than PCA on the average to implement due to its iterative computational nature. These drawbacks make MDS inappropriate for real time browsing or visualization of images.

4 Minimizing Window Overlap

One drawback of a PCA Splat is that similar images are most often partially overlapped which makes the visualization difficult to digest. This problem can be regarded as an optimization of window size and location in order to minimize overlap and hence maximize visibility. The following is the description of the proposed approach.

4.1 Problem formulation

Given a set of windows, the number of windows is *N*. The center coordinates of the windows are denoted as (x_i, y_i), $i = 1, \cdots, N$ and the initial window positions are denoted as (x_i^o, y_i^o), $i = 1, \cdots, N$. The maximum and minimum coordinates of the 2-D screen are $[x_{\min}, x_{\max}, y_{\min}, y_{\max}]$. The window size is represented by its radius r_i, $i = 1, \cdots, N$ for simplicity and the maximum and minimum window size is $r_{\max}$ and $r_{\min}$ in radius. The initial window size is r_i^o, $i = 1, \cdots, N$.

We wish to find a solution that modifies the image windows slightly without deviating too much from the initial locations (which convey mutual similarities). Two factors are taken into account for cost function design. The first is to keep the total area of window overlap as small as possible. The second is to keep the overall deviation from the initial positions on the screen as small as possible.

This represents a trade-off between minimizing overlap and minimizing deviation. To minimize the overlap, one could simply move the windows away from each other but this would simply increase the deviation of the window from the original position. Large deviations are certainly undesirable since the initial positions are important. Without increasing the overall deviation, another way to minimize the overlap is to shrink the window size. Of course, the window size cannot be arbitrary small. Since increasing the window size risks increasing overlap it is disallowed in the optimization process. For this reason the initial window size r_i^o is assumed to be $r_{\max}$, $i = 1, \cdots, N$.

The total cost function is designed as a linear combination of the individual cost function, taking into account each factor mentioned above.

$$J = F(p) + \lambda \cdot S \cdot G(p) \tag{1}$$

where $F(p)$ is the cost function of total overlap and $G(p)$ is the cost function of the total deviation from the original positions and *S* is a scaling factor set to *(N-1)/2* in order to equalize the range of $G(p)$ and $F(p)$. λ is a weight with $\lambda \geq 0$. When λ is zero, the total deviation is ignored in overlap minimization. When λ is less than one, minimizing total overlap is more important than minimizing total deviation, and vice versa for λ is greater than one.

4.2 Cost function design

The cost function of overall overlap is designed as

$$\mathrm{F}(p) = \sum_{i=1}^{N} \sum_{j=i+1}^{N} f(p) \tag{2}$$

$$f(p) = \begin{cases} 1 - e^{-\frac{u^2}{\sigma_f}} & u > 0 \\ 0 & u \leq 0 \end{cases} \tag{3}$$

where $u = r_i + r_j - \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$, u is a measure of overlapping. When $u \le 0$, there is no overlap between the i^{th} window and the j^{th} window, thus the cost is 0. When $u > 0$, there is partial overlap between the i^{th} window and the j^{th} window. When $u = 2 \cdot r_{\max}$, the i^{th} window and the j^{th} window are totally overlapped. σ_f is curvature-controlling factor.

Figure 2 shows the plot of $f(p)$. Note that with an increasing value of u $(u > 0)$, the cost also increases.

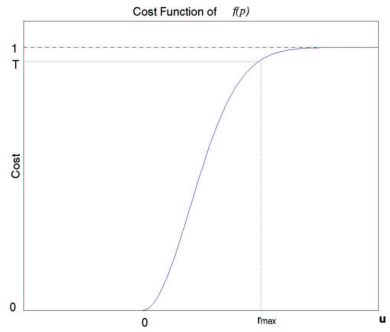


Figure 2. Cost function of overlap function *f(p)*

From Fig. 2, σ_f in Eq. (3) is calculated by setting *T=0.95* when $u = r_{\max}$.

$$\sigma_f = \frac{-u^2}{\ln(1-T)}\Big|_{u=r_{\max}} \tag{4}$$

The cost function of overall deviation is designed as

$$G(p) = \sum_{i=1}^{N} g(p) \tag{5}$$

$$g(p) = 1 - e^{-\frac{v^2}{\sigma_g}} \tag{6}$$

where $v = \sqrt{(x_i - x_i^o)^2 + (y_i - y_i^o)^2}$, v is the deviation of the i^{th} window from its initial position. σ_g is curvature-controlling factor. (x_i, y_i) and (x_i^o, y_i^o) are the optimized and initial center coordinates of the i^{th} window, respectively, $i = 1, \cdots, N$.

Figure 3 shows the plot of *g(p)*. With an increasing value of *v*, the cost of deviation is also increasing.

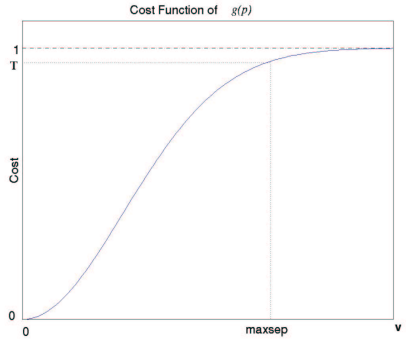


Figure 3. Cost function of function *g(p)*

From Fig. 3, σ_g in Eq. (6) is calculated by setting $T=0.95$ when $v = maxsep$. In our work, *maxsep* is set to be $2 \cdot r_{\max}$.

$$\sigma_g = \frac{-v^2}{\ln(1-T)}\Big|_{v=maxsep} \tag{7}$$

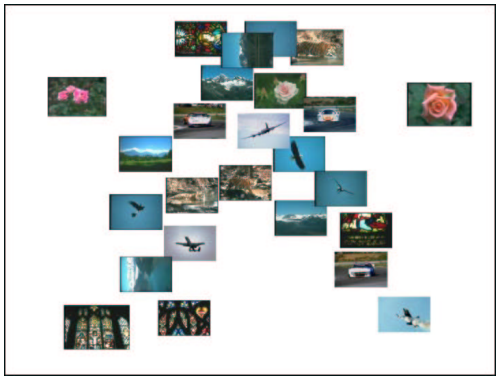
The optimization technique we used was to minimize the total cost J by adjusting the size and positions of the windows until a local minimum is found. It was implemented by an iterative gradient descent procedure. The images are then positioned and sized according to their windows' optimal parameters.

5 Experimental Results

The above technique is a general approach and can be applied to avoid overlapping of various kinds of windows, frames, boxes, images or multimedia icons. When this technique is applied to the PCA Splat, a better content-based visualization can be achieved.

Figure 4 (a) shows the optimized PCA Splat for Fig. 1(a). Clearly, the overlap and crowding is minimized and at the same time the similar images are still close to each other. An improved visualization can even be achieved when this technique is applied to the random layout of Fig. 1(b) (see Figure 4(b)) simply because the overlap is minimized.

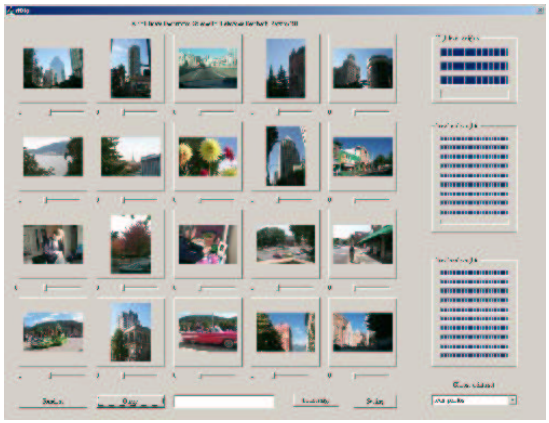
(a) Optimized PCA Splat



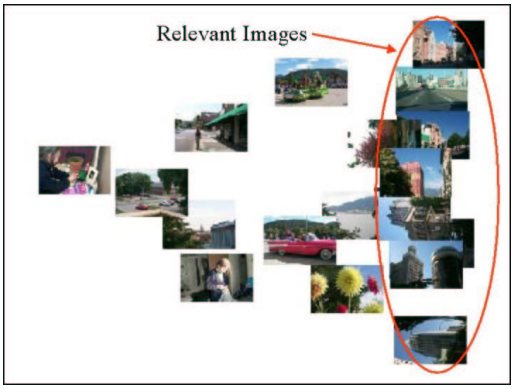
(b) Optimized random display

Figure 4 Optimized displays

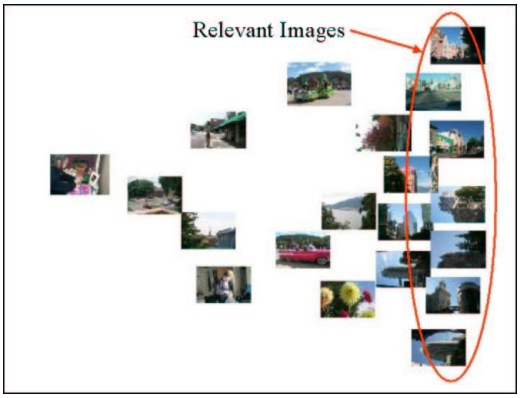
When the proposed optimized technique is applied for content-based image retrieval, a better visualization of the retrieval results can be made, since in addition to rank information (1st-order) one can also see 2nd-order relations (mutual similarities). This would not only provide a better understanding of the query results but also aid the user in forming a new query or provide relevance feedback. An example is shown in Figure 5.



(a) The top 20 retrieved images in our system (ranked from left to right, from top to bottom, the top left 1 is the query)



(b) PCA Splat of the top 20 retrieved images in (a)



(c) Optimized PCA Splat of (b)

Figure 5. An example for Content-Based Image Retrieval

Acknowledgement

This work was supported in part by Mitsubishi Electronic Research Laboratory, Cambridge, MA 02139 and NSF grant CDA 96-24396 and EIA 99-75019.

References

1. C. S. McCamy, H. Marcus, and J. G. Davidson, " A Color-Rendition Chart", *J. Applied Photographic Eng*., Vol. 2, pp. 95-99, Summer 1976.
2. M. Stricker and M. Orengo, "Similarity o Color Images", *Proc. SPIE Storage and Retrieval for Image and Video Databases*, 1995.
3. J. R. Smith and S. F. Chang, "Automated Binary Texture Feature Sets for Image Retrieval", *Proc. IEEE Intl. Conf. Acoust., Speech, and Signal Proc.*, Atlanta, GA, 1996.
4. J. R. Smith and S. F. Chang, "Transform Features for Texture Classification and Discrimination in Large Image Database", *Proc. IEEE Intl. Conf. on Image Proc.*, 1994.
5. S. X. Zhou, Y. Rui and T. S. Huang, "Water-filling algorithm: A novel way for image feature extraction based on edge maps", in *Proc. IEEE Intl. Conf. On Image Proc.*, Japan, 1999.
6. Y. Rubner, "Perceptual metrics for image database navigation", Ph.D. dissertation, Stanford University, 1999.
7. Jolliffe, I.T., *Principal Component Analysis*, Springer-Verlag, New-York, 1986.
8. W. S. Torgeson, *Theory and methods of scaling*, John Wiley and Sons, New York, NY, 1958.