MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Visualization and User-Modeling for Browsing Personal Photo Libraries

Baback Moghaddam, Qi Tian, Neal Lesh, Chia Shen, Thomas S. Huang

TR2004-026 February 2004

Abstract

We present a user-centric system for visualization and layout for content-based image retrieval. Image features (visual and/or semantic) are used to display retrievals as thumbnails in a 2-D spatial layout or ”configuration” which conveys all pair-wise mutual similarities. A graphical optimization technique is used to provide maximally uncluttered and informative layouts. Moreover, a novel subspace feature weighting technique can be used to modify 2-D layouts in a variety of context-dependent ways. An efficient computational technique for subspace weighting and re-estimation leads to a simple user-modeling framework whereby the system can learn to display query results based on layout examples (or relevance feedback) provided by the user. The resulting retrieval, browsing and visualization can adapt to the user's (time-varying) notions of content, context and preferences in style and interactive navigation. Monte Carlo simulations with machine-generated layouts as well as pilot user studies have demonstrated the ability of this framework to model or ”mimic” users, by automatically generating layouts according to their preferences

International Journal of Computer Vision, 56(1/2), pps. 109-130, 2004

Published in: International Journal of Computer Vision, 56 (1/2), pps. 109-130, 2004

Visualization and User-Modeling for Browsing Personal Photo Libraries

Baback Moghaddam[*1] Qi Tian[2], Neal Lesh[1], Chia Shen[1], Thomas S. Huang[2]

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

Abstract

We present a user-centric system for visualization and layout for content-based image retrieval. Image features (visual and/or semantic) are used to display retrievals as thumbnails in a 2-D spatial layout or "configuration" which conveys all pair-wise mutual similarities. A graphical optimization technique is used to provide maximally uncluttered and informative layouts. Moreover, a novel subspace feature weighting technique can be used to modify 2-D layouts in a variety of context-dependent ways. An efficient computational technique for subspace weighting and re-estimation leads to a simple user-modeling framework whereby the system can learn to display query results based on layout examples (or relevance feedback) provided by the user. The resulting retrieval, browsing and visualization can adapt to the user's (time-varying) notions of content, context and preferences in style and interactive navigation. Monte Carlo simulations with machine-generated layouts as well as pilot user studies have demonstrated the ability of this framework to model or "mimic" users, by automatically generating layouts according to their preferences. **Keywords**: CBIR, Visualization, Subspace Analysis, PCA, Estimation.

1 Introduction

Recent advances in technology have made it possible to easily amass large collections of digital recordings of our daily lives. These media offer opportunities for new story sharing experiences beyond the

[1] Corresponding author: Baback Moghaddam, Mitsubishi Electric Research Laboratory (MERL)
201 Broadway, Cambridge MA 02139 USA. Email: baback@merl.com Tel: (617) 621-7524. Fax: (617) 621-7550

conventional digital photo album [1, 2]. The Personal Digital Historian (PDH) project is an ongoing effort to help people construct, organize, navigate, and share digital collections in an interactive multi-person conversational setting [3, 4]. The research in PDH is guided by the following principles:

(1) The display device should enable natural face-to-face conversation – not forcing everyone to face in the same direction (desktop) or at their own separate displays (hand-held devices).

(2) The physical sharing device must be convenient and customary to use – helping to make the computer disappear.

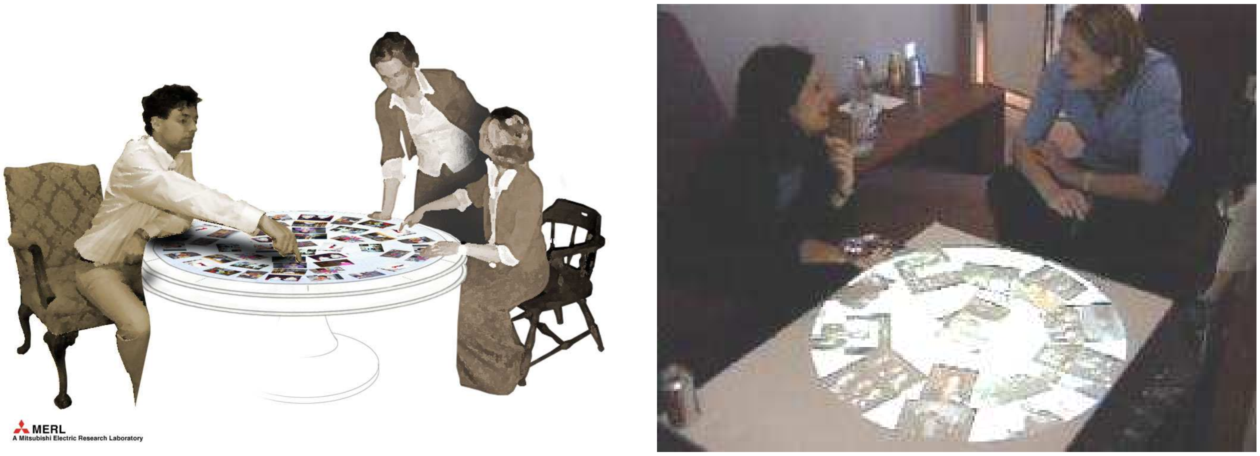
(3) Easy and fun to use across generations of users – minimizing time spent typing or formulating queries.

(4) Enabling interactive and exploratory storytelling – blending authoring and presentation.

Current software and hardware do not meet our requirements. Most existing software in this area provides users with either powerful query methods or authoring tools. In the former case, the users can repeatedly query their collections of digital content to retrieve information to show someone [5]. In the latter case, a user experienced in the use of the authoring tool can carefully craft a story out of their digital content to show or send to someone at a later time. Furthermore, current hardware is also lacking. Desktop computers are not suitably designed for group, face-to-face conversation in a social setting, while handheld story-telling devices have limited screen sizes and can be used only by a small number of people at once. The objective of the PDH project is to take a step beyond.

The goal of PDH is to provide a new digital content user interface and management system enabling face-to-face causal exploration and visualization of digital contents. Unlike conventional desktop user interface, PDH is intended for multi-user collaborative applications on single display groupware. PDH enables casual and exploratory retrieval, interaction with and visualization of digital contents.

We designed our system to work on a touch-sensitive, circular table-top display [6], as shown in Figure 1. The physical PDH table, we use a standard tabletop with a top projection (either ceiling mounted or tripod mounted) that displays on a standard whiteboard as shown in the right image of Figure 1. We used two Mimio [7] styluses as the input devices for first set of user experiments. The layout of the entire table-top display, as shown in Figure 2, consists of (1) a large story-space area encompassing most of the table-top until the perimeter, and (2) one or more narrow arched control panels. Currently, the present PDH table is implemented using our DiamondSpin (www.merl.com/projects/diamondspin) circular table Java toolkit. DiamondSpin is intended for multi-user collaborative applications [4, 6, 8].



(a) (b)

Figure 1. PDH Table (a) An artistic rendering of the PDH table (designed by Ryan Bardsley, Tixel HCI www.tixel.net) and (b) the physical PDH table

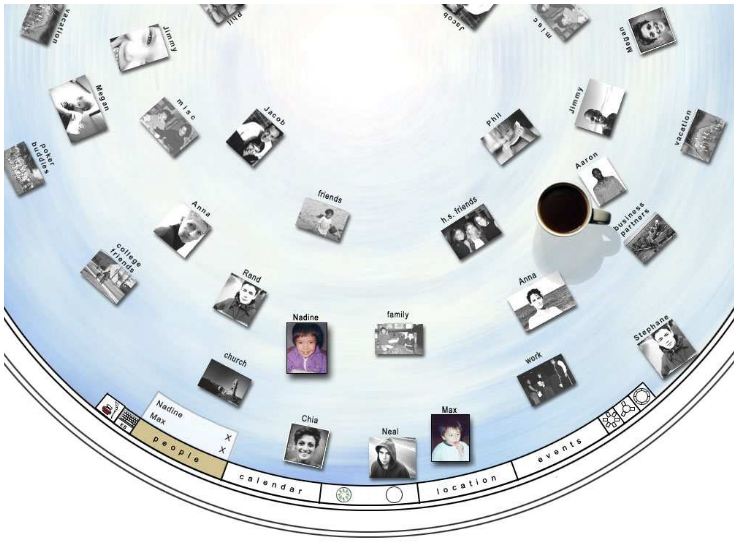


Figure 2: Close-up of “people-view” on the PDH table.

The conceptual model of PDH was to focus on developing content organization and retrieval metaphors that can easily understandable by users without distracting from the conversation. We adopted a model of organizing the materials using the four questions essential to storytelling: who, when, where, and what (the four Ws). We do not currently support why, which is also useful for storytelling. Control panels located on the perimeter of the table contain buttons labeled “people”, “calendar”, “location”, and “events”, corresponding to these four questions. When a user presses the “location” button, for example, the display on the table changes to show a map of the world. Every picture in the database that is annotated with a location will appear as a tiny thumbnail at its location. The user can pan and zoom in on the map to a region of interest, which increase the size of the thumbnails. Similarly, by pressing one of the other three buttons, the user can cause the pictures to be organized by the time they were taken along a linear straight timeline, the people they contain, or the event keywords with which the pictures were annotated. We assume the pictures are partially annotated. Figure 3 shows an example of navigation of personal photo album by the four Ws model. Adopting this model allows users to think of their documents in terms of how they would like to record them as part of their history collection, not necessarily in a specific hierarchical structure. The user can make selections among the four Ws and PDH will automatically combine them to form rich Boolean queries implicitly for the user [3, 4, 6, 8]

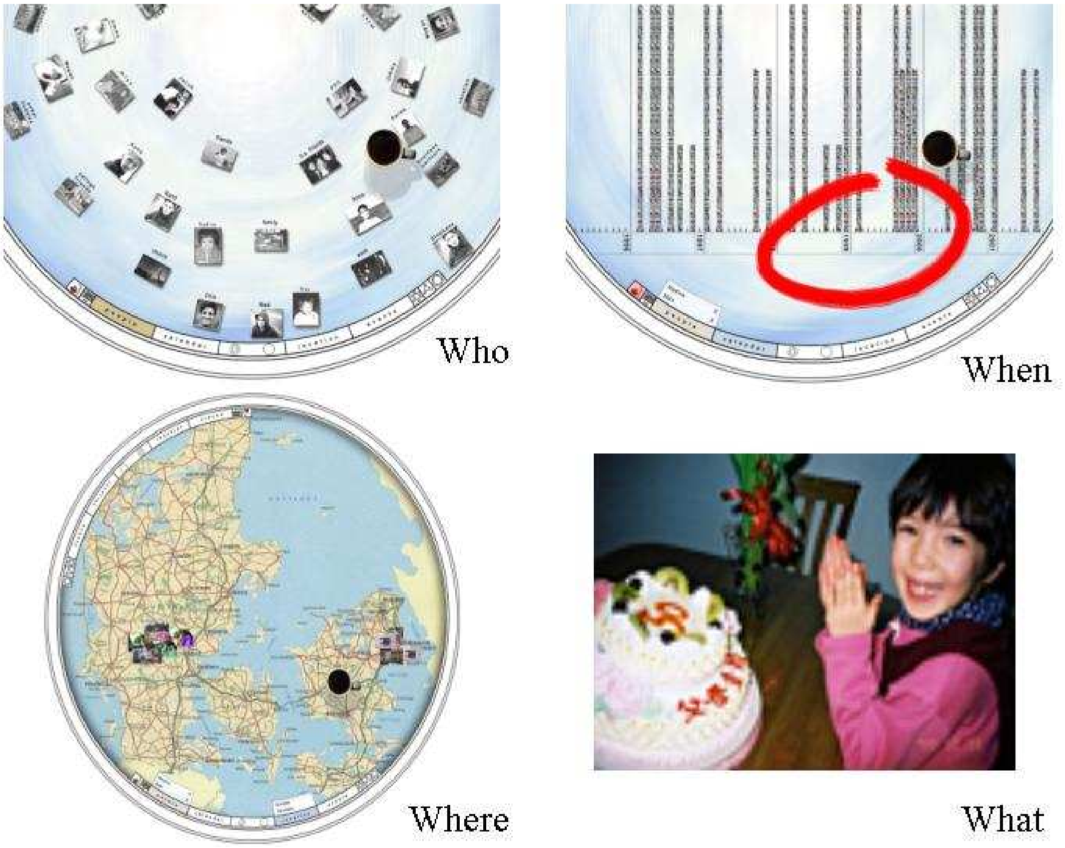


Figure 3: An example of navigation by the four Ws model (Who, When, Where, What)

The PDH project combines and extends research in largely two areas: (1) human-computer interaction (HCI) and interface (the design of the shared-display devices, user interface for story-telling and on-line authoring, and story-listening); (2) content-based information visualization, presentation and retrieval (user-guided image layout, data mining and summarization). Previous work in [3, 4, 6, 8] focused on the HCI and interface design issues of the first research area above. In this paper, we present our work for the second research area above. The proposed visualization and layout algorithms [9-12] that can enhance informal storytelling using personal digital data such as photos, audio and video in a face-to-face social setting.

Due to the semantic gap [13], visualization is very important for user to navigate the complex query space. New visualization tools are required to allow for user-dependent and goal-dependent choices about what to display and how to provide feedback. The query result has an inherent display dimension that is often ignored. Most methods display images in a 1-D list in order of decreasing similarity to the query images. Enhancing the visualization of the query results is, however, a valuable tool in helping the user-navigating query space. Recently, other researchers have also explored towards content-based visualization [14-18]. A common observation in [14-18] is that the images are displayed in 2-D or 3-D space from the projection of the high-dimensional feature spaces. Images are placed in such a way that distances between images in 2-D or 3-D reflect their distances in the high-dimensional feature space. In [14, 15], the users can view large sets of images in 2-D or 3-D space and user navigation is allowed. In [15-17], the system allows user interaction on image location and forming new groups. In [16-17], users can manipulate the projected distances between images and learn from such a display. As is apparent from the query space framework, there is an abundance of information available for display. Our work in [9-12] under the context of PDH shares many common features with the related work in [14-18]. However, learning mechanism from display is not implemented in [14] and 3D MARS [15] is an extension to our work [9, 10] from 2-D to 3-D space. Our system differs from the work in [18] is that we adopted different mapping methods. Our work shares some features with the work by Santini & Jain [16-17] except that our PDH system is currently being incorporated into a much broader system for computer-human guided navigating, browsing, archiving, and interactive story-telling with large photo libraries. The part of this system described in the remainder of this paper is, however, specifically geared towards adaptive user-modeling and relevance estimation and based primarily on visual features as opposed to semantic annotation as in [16-17].

The rest of paper is organized as follows. In Section 2, we present designs for optimal (uncluttered) visualization and layout of images (or iconic data in general) in a 2-D display space for content-based image retrieval [9, 10]. In Section 3, we further provide a mathematical framework for user-modeling, which adapts and mimics the user’s (possibly changing) preferences and style for interaction, visualization and navigation [11, 12]. Monte Carlo simulations in Section 4 and pilot user studies in Section 5 have demonstrated the ability of our framework to model or “mimic” users, by automatically generating layouts according to user’s preference. Finally discussion is given in Section 6 and the conclusion in Section 7.

2 Visualization

With the advances in technology to capture, generate, transmit and store large amounts of digital imagery and video, research in content-based image retrieval (CBIR) has gained increasing attention. In CBIR, images are indexed by their visual contents such as color, texture, etc. Many research efforts have addressed how to extract these low level features [19-21], evaluate distance metrics [22, 23] for similarity measures and look for efficient searching schemes [24, 25].

In this section, we present a user-centric system for visualization and layout for content-based image retrieval. Image features (visual and/or semantic) are used to display retrievals as thumbnails in a 2-D spatial layout or "configuration" which conveys pair-wise mutual similarities. A graphical optimization technique is used to provide maximally uncluttered and informative layouts. We should note that one physical instantiation of the PDH table is that of a roundtable, for which we have in fact experimented with polar coordinate conformal mappings for converting traditional rectangular display screens. However, in the remainder of this paper, for purposes of ease of illustration and clarity, all layouts and visualizations are shown on rectangular displays only.

2.1 Traditional Interfaces

The purpose of automatic content-based visualization is augmenting the user's understanding of large information spaces that cannot be perceived by traditional sequential display (*e.g.* by rank order of visual similarities). The standard and commercially prevalent image management and browsing tools currently available primarily use tiled sequential displays – *i.e.* essentially a simple 1-D similarity-based visualization.

However, the user quite often can benefit by having a global view of a working subset of retrieved images in a way that reflects the relations between *all pairs* of images – i.e., N^2 measurements as opposed to only N. Moreover, even a narrow view of one's immediate surroundings defines "context" and can offer an indication on how to explore the dataset. The wider this "visible" horizon, the more efficient the new query will be formed. In [18], Rubner proposed a 2-D display technique based on multi-dimensional scaling (MDS) [26]. A global 2D view of the images is achieved that reflects the mutual similarities among the retrieved images. MDS is a nonlinear transformation that minimizes the stress between high dimensional feature space and low dimensional display space. However, MDS is rotation invariant, non-repeatable (non-unique), and often slow to implement. Most critically, MDS (as well as some of the other leading nonlinear dimensionality reduction methods) provide high-to-low-dimensional projection operators that are not analytic or functional in form, but are rather defined on a point-by-point basis for each given dataset. This makes it very difficult to project a new dataset in a functionally consistent way (without having to build a post-hoc projection or interpolation function for the forward mapping each time). We feel that these drawbacks make MDS (and other nonlinear methods) an unattractive option for real time browsing and visualization of high-dimensional data such as images.

2.2 Improved Layout & Visualization

We propose an alternative 2-D display scheme based on Principle Component Analysis (PCA) [27]. Moreover, a novel window display optimization technique is proposed which provides a more perceptually intuitive, visually uncluttered and informative visualization of the retrieved images.

Traditional image retrieval systems display the returned images as a list, sorted by decreasing similarity to the query. The traditional display has one major drawback. The images are ranked by similarity to the query, and relevant images (as for example used in a relevance feedback scenario) can appear at separate and distant locations in the list. We propose an alternative technique to MDS in [26] that displays mutual similarities on a 2-D screen based on visual features extracted from images. The retrieved images are displayed not only in ranked order of similarity from the query but also according to their mutual similarities, so that similar images are grouped together rather than being scattered along the entire returned 1-D list.

2.3 Visual Features

We will first describe the low-level visual feature extraction used in our system. There are three visual features used in our system: color moments [19], wavelet-based texture [20], and water-filling edge-based structure feature [21]. The color space we use is HSV because of its de-correlated coordinates and its perceptual uniformity [19]. We extract the first three moments (mean, standard deviation and skewness) from the three-color channels and therefore have a color feature vector of length $3 \times 3 = 9$.

For wavelet-based texture, the original image is fed into a wavelet filter bank and is decomposed into 10 de-correlated sub-bands. Each sub-band captures the characteristics of a certain scale and orientation of the original image. For each sub-band, we extract the standard deviation of the wavelet coefficients and therefore have a texture feature vector of length 10.

For water-filling edge-based structure feature vector, we first pass the original images through an edge detector to generate their corresponding edge map. We extract eighteen (18) elements from the edge maps, including *max fill time, max fork count, etc.* For a complete description of this edge feature vector, interested readers are referred to [21].

2.4 PCA Splats

To create such a 2-D layout, Principle Component Analysis (PCA) [27] is first performed on the retrieved images to project the images from the high dimensional feature space to the 2-D screen. Image thumbnails are placed on the screen so that the screen distances reflect as closely as possible the similarities between the images. If the computed similarities from the high dimensional feature space agree with our perception, and if the resulting feature dimension reduction preserves these similarities reasonably well, then the resulting spatial display should be informative and useful.

In our experiments, the 37 visual features (9 color moments, 10 wavelet moments and 18 water-filling features) are pre-extracted from the image database and stored off-line. Any 37-dimensional feature vector for an image, when taken in context with other images, can be projected on to the 2-D $\{x, y\}$ screen based on the 1st two principal components normalized by the respective eigenvalues. Such a layout is denoted as a PCA Splat. We implemented both linear and non-linear projection methods using PCA and Kruskal's algorithm [26]. The projection using the non-linear method such as the Kruskal's algorithm is an iterative procedure, slow to converge and converged to the local minima. Therefore the convergence largely depends on the initial starting point and cannot be repeatable. On the contrary PCA has several advantages over nonlinear methods like MDS. It is a fast, efficient and unique linear transformation that achieves the maximum distance preservation from the original high dimensional feature space to 2-D space among all possible linear transformations [27]. The fact that it fails to model nonlinear mappings (which MDS succeeds at) is in our opinion a minor compromise given the advantages of real-time, repeatable and mathematically tractable linear projections.

Let us consider a scenario of a typical image-retrieval engine at work in which an actual user is providing relevance feedback for the purposes of query refinement. Figure 4 shows an example of the retrieved images by the system (which resembles most traditional browsers in its 1D tile-based layout). The database is a collection of 534 images. The 1st image (building) is the query. The other 9 relevant images are ranked in 2nd, 3rd, 4th, 5th, 9th, 10th, 17th, 19th and 20th places, respectively.

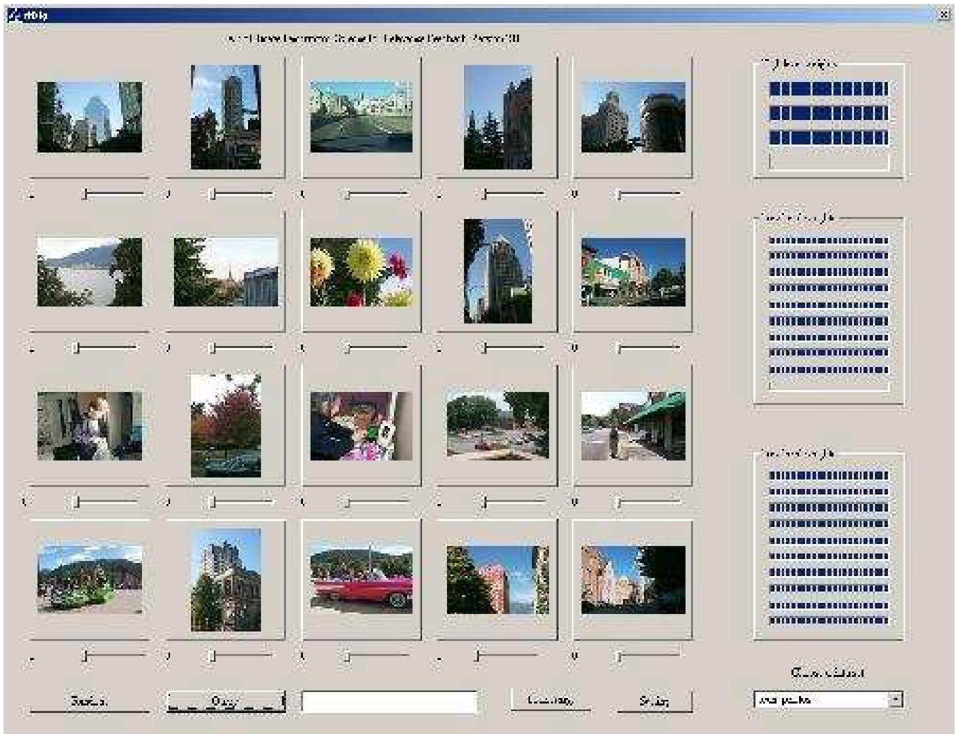


Figure 4. Top 20 retrieved images (ranked top to bottom and left to right; query is shown first in the list)

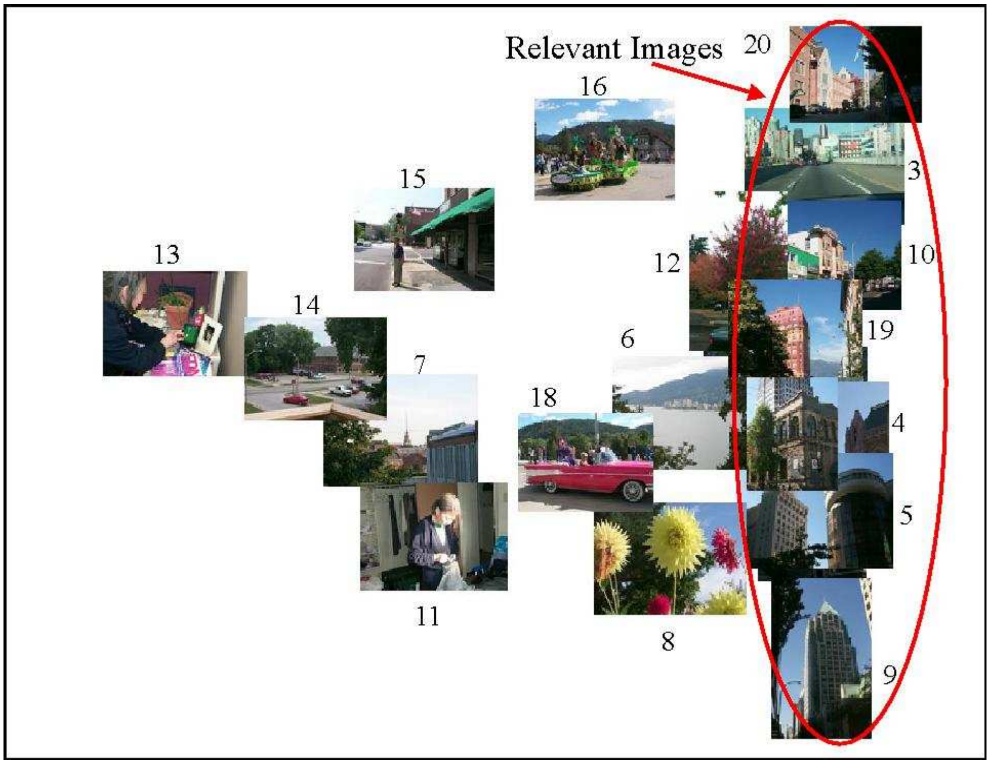


Figure 5. PCA Splat of top 20 retrieved images in Figure 3

Figure 5 shows an example of a PCA Splat for the top 20 retrieved images shown in Figure 4[2]. In addition to visualization by layout, in this particular example, the sizes (alternatively contrast) of the images are determined by their visual similarity to the query. The higher the rank, the larger the size (or higher the contrast). There is also a number next to each image in Figure 5 indicating its corresponding rank in Figure 4. The view of query image, i.e., the top left one in Figure 4, is blocked by the images ranked 19th, 4th, and 17th in Figure 5. A better view is achieved in Figure 8 after display optimization.

Clearly the relevant images are now better clustered in this new layout as opposed to being dispersed along the tiled 1-D display in Figure 4. Additionally, PCA Splats convey N^2 mutual distance measures relating all pair-wise similarities between images while the ranked 1-D display in Figure 4 provides only *N*.

2.5 Display Optimization

However, one drawback of PCA Splat is that some images can be partially or totally overlapped which makes it difficult to view all the images at the same time. The overlap will be even worse when the number of retrieved images becomes larger, e.g. larger than 50. To solve the overlapping problem between the retrieved images, a novel optimized technique is proposed in this section.

Given the sets of the retrieved images and their corresponding sizes and positions, our optimizer tries to find a solution that places the images at the appropriate positions while deviating as little as possible from their initial PCA Splat positions. Assume the number of images is *N*. The image positions are represented by their center coordinates (x_i, y_i), $i = 1,\cdots,N$, and the initial image positions are denoted as (x_i^o, y_i^o), $i = 1,\cdots,N$. The minimum and maximum coordinates of the 2-D screen are $[x_{\min}, x_{\max}, y_{\min}, y_{\max}]$. The image size is represented by its radius r_i for simplicity, $i = 1,\cdots,N$ and the maximum and minimum image size is $r_{\max}$ and $r_{\min}$ in radius, respectively. The initial image size is r_i^o, $i = 1,\cdots,N$.

To minimize the overlap, one can move the images away from each other to decrease the overlap between images, but this will increase the deviation of the images from their initial positions. Large deviation is certainly undesirable because the initial positions provide important information about mutual similarities between images. So there is a trade-off problem between minimizing overlap and minimizing deviation.

[2] High-resolution version of all the figures in this paper can be found at http://www.ifp.uiuc.edu/~qitian/ijcv/

Without increasing the overall deviation, an alternative way to minimize the overlap is to simply shrink the image size as needed, down to a minimum size limit. The image size will not be increased in the optimization process because this will always increase the overlap. For this reason, the initial image size r_i^o is assumed to be $r_{\max}$.

The total cost function is designed as a linear combination of the individual cost functions taking into account two factors. The first factor is to keep the overall overlap between the images on the screen as small as possible. The second factor is to keep the overall deviation from the initial position as small as possible.

$$J = F(p) + \lambda \cdot S \cdot G(p) \qquad (1)$$

where $F(p)$ is the cost function of the overall overlap and $G(p)$ is the cost function of the overall deviation from the initial image positions, S is a scaling factor which brings the range of $G(p)$ to the same range of $F(p)$ and S is chosen to be $(N-1)/2$. λ is a weight and $\lambda \geq 0$. When λ is zero, the deviation of images is not considered in overlapping minimization. When λ is less than one, minimizing overall overlap is more important than minimizing overall deviation, and vice versa for λ is greater than one.

The cost function of overall overlap is designed as

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

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

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

Figure 6 shows the plot of $f(p)$. With the increasing value of u $(u > 0)$, the cost of overlap is also increasing.

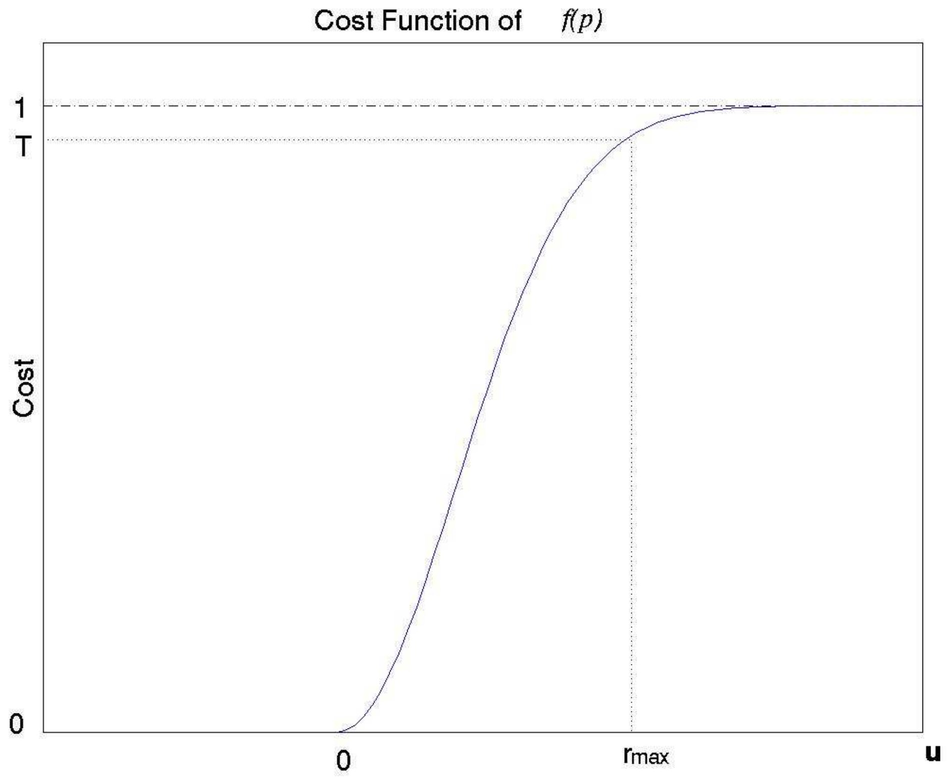


Figure 6. Cost function of overlap function *f(p)*

From Fig. 6, 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 measure of deviation of the ith image from its initial position. σ_g is a curvature-controlling factor. (x_i, y_i) and (x_i^o, y_i^o) are the optimized and initial center coordinates of the ith image, respectively, $i = 1, \cdots, N$.

Figure 7 shows the plot of *g(p)*. With the increasing value of *v*, the cost of deviation is also increasing.

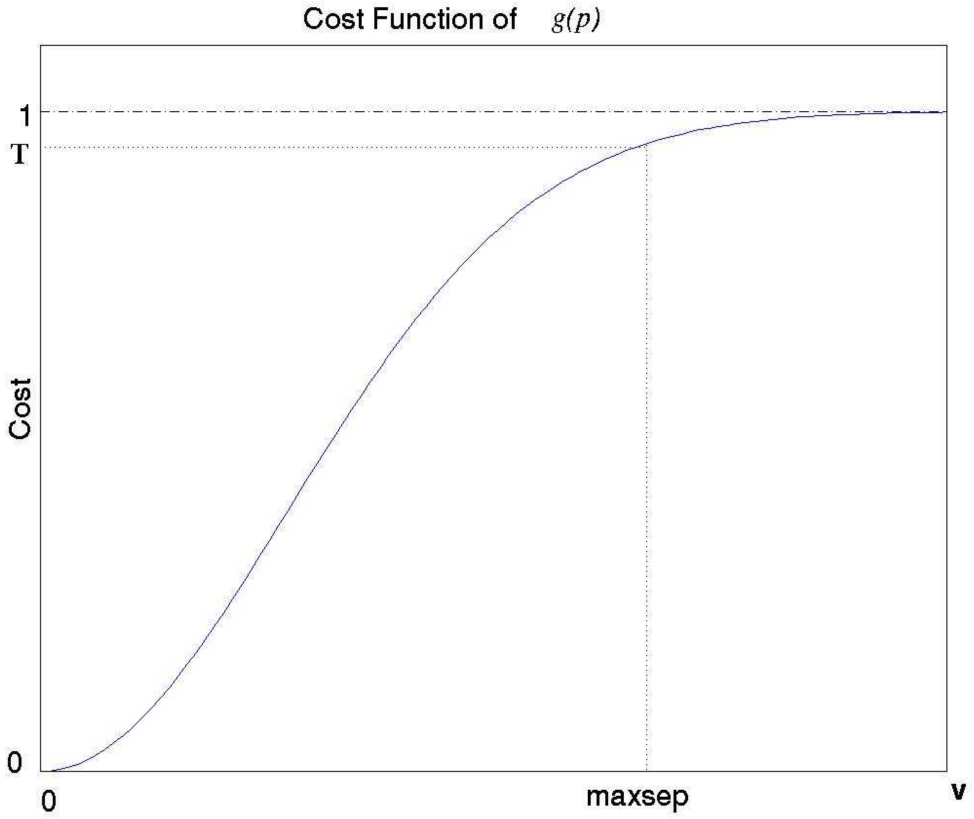


Figure 7. Cost function of function *g(p)*

From Fig. 7, σ_g in Eq. (6) is calculated by setting *T=0.95* when $v = maxsep$. In our work, *maxsep* is set to be $2 \cdot r_{\max}$.

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

The optimization process is to minimize the total cost *J* by finding a (locally) optimal set of size and image positions. The nonlinear optimization method was implemented by an iterative gradient descent method (with line search). Once converged, the images will be redisplayed based on the new optimized sizes and positions.

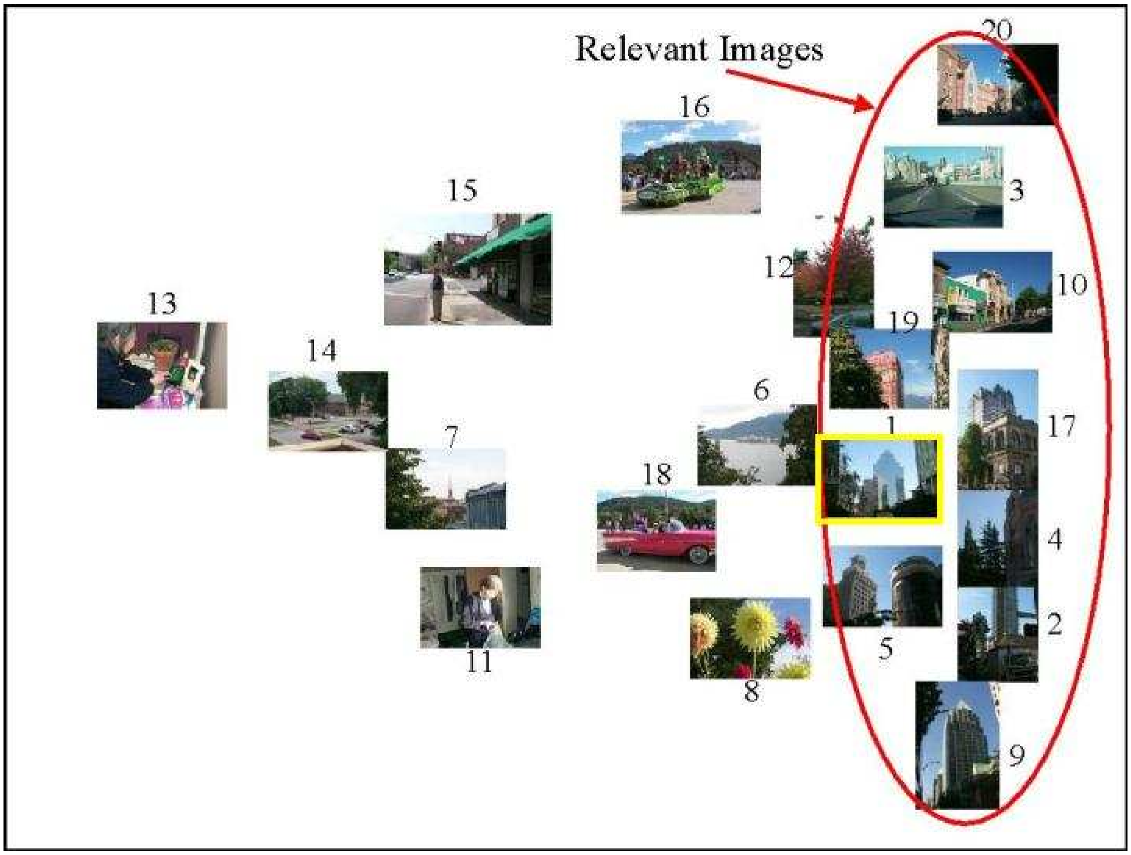


Figure 8. Optimized PCA Splat of Figure 4

Figure 8 shows the optimized PCA Splats for Fig. 4. The image with a yellow frame is the query image in Figure 4. Clearly, the overlap is minimized while the relevant images are still close to each other to allow a global view. With such a display, the user can see the relations between the images, better understand how the query performed, and subsequently formulate future queries more naturally. Additionally, attributes such as contrast and brightness can be used to convey rank. We note that this additional visual aid is essentially a 3rd dimension of information display. For example, images with higher rank could be displayed with larger size or increased brightness to make them stand out from the rest of the layout. An interesting example is to display "time" or "timeliness" by associating the size or brightness with how long ago the picture was taken, thus images from the "past" would appear smaller or dimmer than those taken recently. A full discussion of the resulting enhanced layouts is deferred to future work.

Also we should point out that despite our ability to "clean-up" layouts for maximal visibility with the optimizer we have designed, all subsequent figures in this paper show Splats *without* any overlap minimization since for illustrating (as well as comparing) the accuracy of the estimation results in subsequent sections the absolute position was necessary and important.

3 Context and User-Modeling

Image content and "meaning" is ultimately based on semantics. The user's notion of content is a high-level concept, which is quite often removed by many layers of abstraction from simple low-level visual features. Even near-exhaustive semantic (keyword) annotations can never fully capture context-dependent notions of content. The same image can "mean" a number of different things depending on the particular circumstance. The visualization and browsing operation should be aware of which feature (visual and/or semantic) are relevant to the user's current focus (or working set) and which should be ignored. In the space of all possible features for an image, this problem can be formulated as a subspace identification or feature weighting technique that is described fully in Section 3.1

3.1 Estimation of Feature Weights

By user-modeling or "context awareness" we mean that our system must be constantly aware of and adapting to the changing concepts and preferences of the user. A typical example of this human-computer synergy is having the system learn from a user-generated layout in order to visualize new examples based on identified relevant/irrelevant features. In other words, design smart browsers that "mimic" the user, and over-time, adapt to their style or preference for browsing and query display. Given information from the layout, e.g., positions and mutual distances between images, a novel feature weight estimation scheme, noted as $\boldsymbol{\alpha}$-estimation is proposed, where $\boldsymbol{\alpha}$ is a weighting vector for feature e.g., color, texture and structure (and semantic keywords).

We now describe the subspace estimation of $\boldsymbol{\alpha}$ for visual features only, e.g., color, texture, and structure, although it should be understood that the features could include visual, audio and semantic features or any hybrid combination thereof.

In this case, the weighting vector is $\boldsymbol{\alpha} = \{\alpha_{\mathrm{c}}, \alpha_t, \alpha_s\}^T$, where α_c is the weight for color, α_t is the weight for texture, and α_s is the weight for structure. The lengths of color, texture and structure features are L_c, L_t, and L_s, respectively. The number of images in the preferred clustering is *N*, and $\mathbf{X}_c$ is a $L_c \times N$ matrix where the *i*th column is the color feature vector of the *i*th image, $i = 1, \cdots, N$, $\mathbf{X}_t$ is the $L_t \times N$ matrix, the *i*th column is the texture feature vector of the *i*th image, $i = 1, \cdots, N$, and $\mathbf{X}_\mathbf{s}$ is the

$L_s \times N$ matrix, the *i*th column is the structure feature vector of the *i*th image, $i = 1, \cdots, N$. The distance, for example Euclidean-based between the *i*th image and the *j*th image, for $i, j = 1, \cdots, N$, in the preferred clustering (distance in 2-D space) is d_{ij} . These weights α_c , α_t , α_s are constrained such that they always sum to 1.

We then define an energy term to minimize with an *Lp* norm (with $p = 2$). This cost function is defined in Equation (8). It is a nonnegative quantity that indicates how well mutual distances are preserved in going from the original high dimensional feature space to 2-D space. Note that this cost function is similar to MDS stress, but unlike MDS, the minimization is seeking the optimal feature weights $\boldsymbol{\alpha}$. Moreover, the low-dimensional projections in this case are already known. The optimal weighting parameter recovered is then used to weight original feature-vectors before applying a PCA Splat which will result in the desired layout.

$$J = \sum_{i=1}^{N}\sum_{j=1}^{N}\{d_{ij}{}^{p} - \sum_{k=1}^{L_c}\alpha_c{}^{p}\,|\,\mathbf{X}_{c(i)}^{(k)} - \mathbf{X}_{c(j)}^{(k)}\,|^{p} - \sum_{k=1}^{L_t}\alpha_t{}^{p}\,|\,\mathbf{X}_{t(i)}^{(k)} - \mathbf{X}_{t(j)}^{(k)}\,|^{p} - \sum_{k=1}^{L_s}\alpha_s{}^{p}\,|\,\mathbf{X}_{s(i)}^{(k)} - \mathbf{X}_{s(j)}^{(k)}\,|^{p}\}^{2} \quad (8)$$

The global minimum of this cost function corresponding to the optimal weight parameter $\boldsymbol{\alpha}$, is easily obtained using a constrained (non-negative) least-squares as shown in Appendix A. We note that there is an alternative approach to estimating the subspace weighting vector $\boldsymbol{\alpha}$ in the sense of *minimum deviation*, which we have called "deviation-based" $\boldsymbol{\alpha}$-estimation. The cost function in this case is defined as follows:

$$J = \sum_{i=1}^{N}|\,p^{(i)}(x,y) - \hat{p}^{(i)}(x,y)\,|^{2} \quad (9)$$

where $p^{(i)}(x,y)$ and $\hat{p}^{(i)}(x,y)$ are the original and projected 2-D locations of the i^{th} image, respectively. This formulation is a more direct approach to estimation since it deals with the final position of the images in the layout. Unfortunately, however, this approach requires the simultaneous estimation of both the weight vectors as well as the projection basis and consequently requires less-accurate iterative *re-estimation* techniques (as opposed to more robust closed-form solutions possible with (8)). A full derivation of the solution for our deviation-based $\boldsymbol{\alpha}$ estimation is shown in Appendix B. In all the experiments reported in this paper, we used the stress-based method (8) for $\boldsymbol{\alpha}$ estimation.

We note that in principle it is possible to use a single weight for each dimension of the feature vector. However this would lead to a poorly determined estimation problem since it is unlikely (and/or undesirable) to have that many sample images from which to estimate all individual weights. Even with plenty of examples (an over-determined system), chances are that the estimated weights would generalize poorly to a new set of images -- this is the same principle used in a modeling or regression problem where the order of the model or number of free parameters should be less than the number of available observations. Therefore, the key reason for having fewer weights is to avoid the problem of *over-fitting* and the subsequent poor generalization on new data. In this respect, the less weights (or more subspace "groupings") there are, the better the generalization performance. Since the origin of all 37 features is basically from 3 different (independent) visual attributes: color, texture and structure, it seems prudent to use 3 weights corresponding to these 3 subspaces. Furthermore, this number is sufficiently small to almost guarantee that we will always have enough images in one layout from which to estimate these 3 weights.

Figure 9 shows a simple user layout where 3 car images are clustered together despite their different colors. The same is performed with 3 flower images (despite their texture/structure). These two clusters maintain a sizeable separation thus suggesting two separate concept classes implicit by the user's placement. Specifically, in this layout the user is clearly concerned with the distinction between *car* and *flower* regardless of color or other possible visual attributes.

Applying the $\boldsymbol{\alpha}$-estimation algorithm to Figure 9, the feature weights learned from this layout are $\alpha_c = 0.3729$, $\alpha_t = 0.5269$ and $\alpha_s = 0.1002$. This shows that the most important feature in this case is texture and not color, which is in accord with the concepts of car *vs*. flower as graphically indicated by the user in Figure 9.

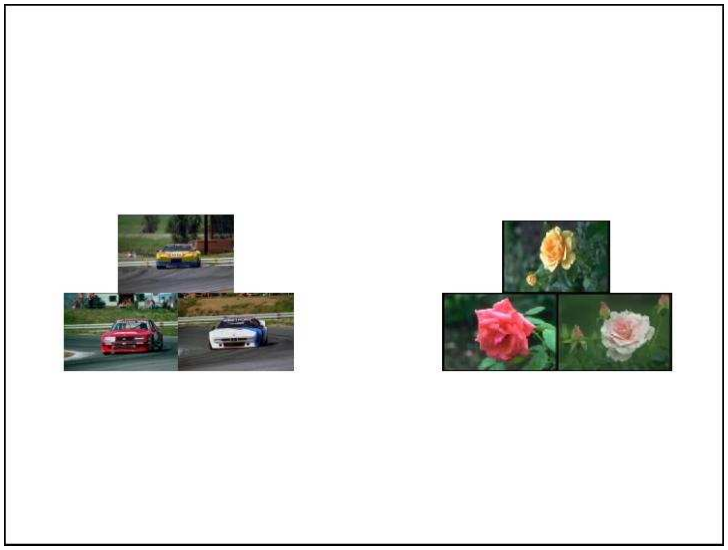
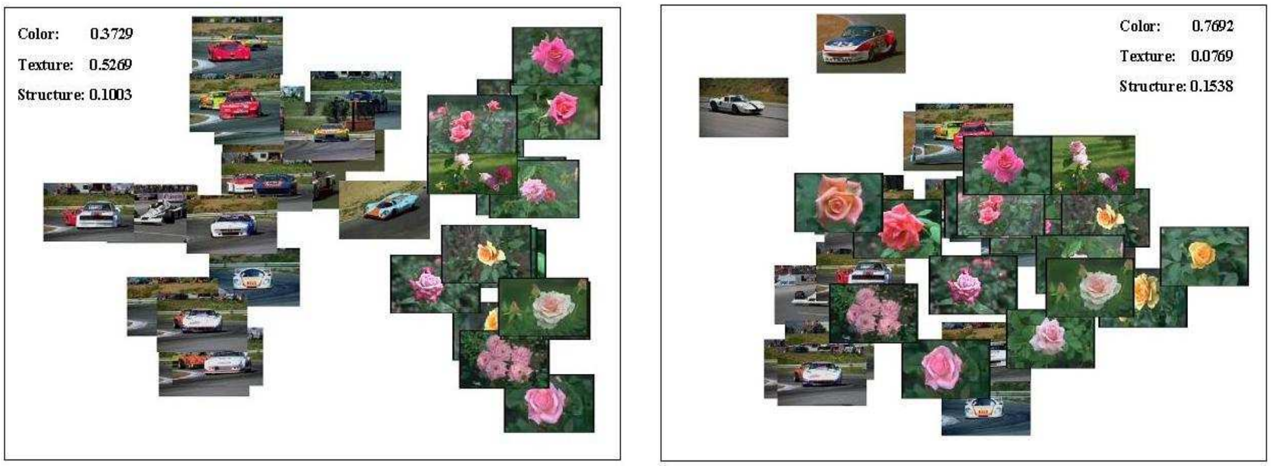


Figure 9. An example of a user-guided layout

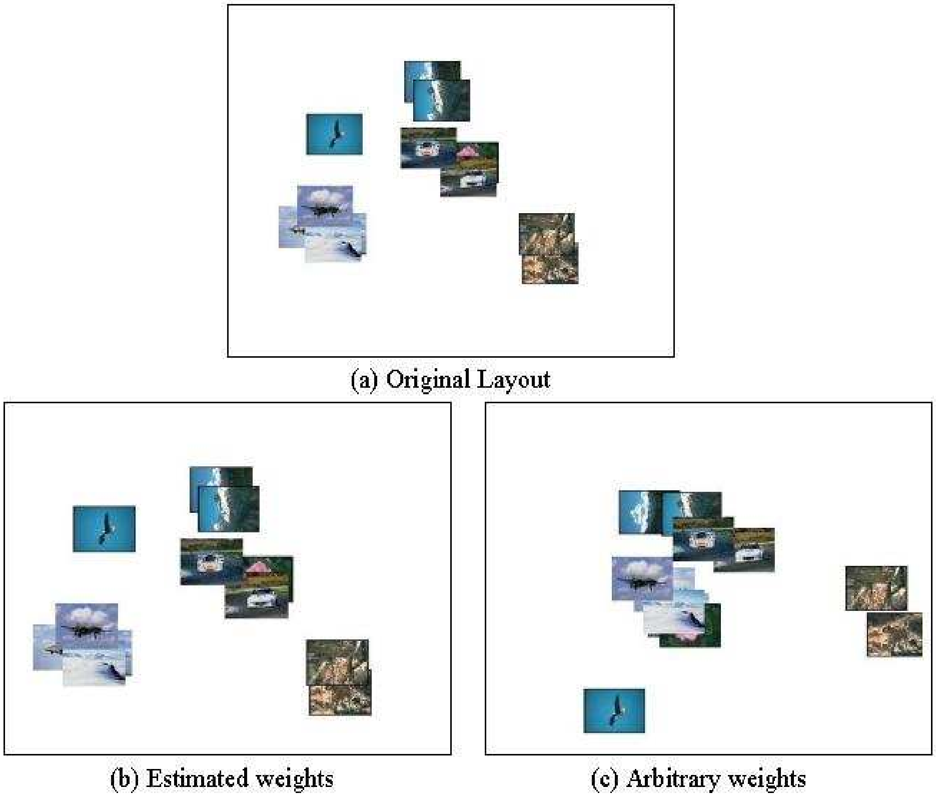


(a) (b)

Figure 10. PCA Splat on a larger set of images using (a) estimated weights (b) arbitrary weights

Now that we have the learned feature weights (or modeled the user) what can we do with them? Figure 10 shows an example of a typical application: automatic layout of a larger (more complete data set) in the style indicated by the user. Fig. 10(a) shows the PCA splat using the learned feature weight for 18 cars and 19 flowers. It is obvious that the PCA splat using the estimated weights captures the essence of the configuration layout in Figure 9. Figure 10(b) shows a PCA splat of the same images but with a randomly generated $\boldsymbol{\alpha}$, denoting an arbitrary but coherent 2-D layout, which in this case, favors color ($\alpha_c = 0.7629$). This comparison reveals that proper feature weighting is an important factor in generating the user-desired and sensible layouts. We should point out that a random $\boldsymbol{\alpha}$ does not generate a random layout, but rather one that is still coherent, displaying consistent groupings or clustering. Here we have used such "random" layouts as substitutes for alternative (arbitrary) layouts that are nevertheless "valid" (differing only in the relative contribution of the 3 features to the final design of the layout). Given the difficulty of obtaining hundreds (let alone thousands) of real user-layouts that are needed for more complete statistical tests (such as those in the next section), random $\boldsymbol{\alpha}$ layouts are the only conceivable way of "simulating" a layout by a real user in accordance with "familiar" visual criteria such as color, texture or structure.

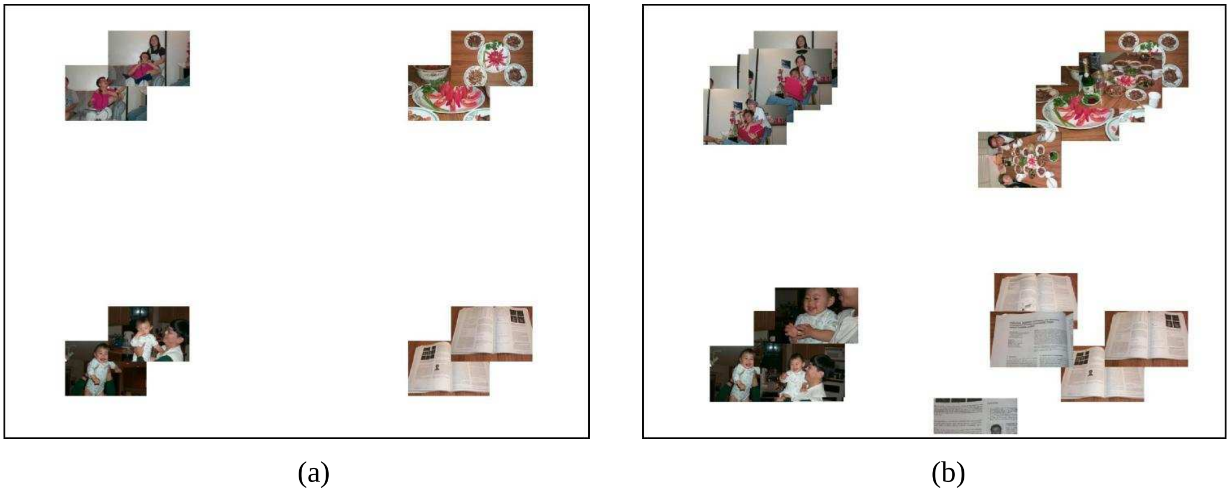
Figure 11(a) shows an example of another layout. Figure 11(b) shows the corresponding computer-generated layout of the same images with their high dimensional feature vectors weighted by the estimated$\boldsymbol{\alpha}$, which is recovered solely from the 2-D configuration of Figure 11(a). In this instance the reconstruction of the layout is near perfect thus demonstrating that our high-dimensional subspace feature weights can in fact be recovered from pure 2-D information. For comparison, Figure 11(c) shows the PCA Splat of the same images with their high dimensional feature vectors weighted by a random $\boldsymbol{\alpha}$.


(a) Original Layout

(b) Estimated weights

(c) Arbitrary weights

Figure 11. (a) An example layout. Computer-generated layout based on (b) reconstruction using learned feature weights, and (c) the control (arbitrary weights)



(a) (b)

Figure 12: User-modeling for automatic layout. (a) a user-guided layout: (b) computer layout for larger set of photos (4 classes and 2 photos from each class)

Figure 12 shows another example of user-guided layout. Assume that the user is describing her family story to a friend. In order not to disrupt the conversation flow, she only lays out a few photos from her personal photo collections and expects the computer to generate a similar and consistent layout for a larger set of images from the same collection. Figure 12(b) shows the computer-generated layout based on the learned feature weights from the configuration of Fig. 12(a). The computer-generated layout is achieved using the α-estimation scheme and post-linear, e.g., affine transform or non-linear transformations. Only the 37 visual features (9 color moments [5], 10 wavelet moments [13] and 18 water-filling features [14]) were used for this PCA Splat. Clearly the computer-generated layout is similar to the user layout with the visually similar images positioned at the user-indicated locations. We should add that in this example no semantic features (keywords) were used, but it is clear that their addition would only enhance such a layout.[3]

4 Statistical Analysis

Given the lack of sufficiently large (and willing) human subjects, we undertook a Monte Carlo approach to testing our user-modeling and estimation method. Thereby simulating 1000 computer generated layouts (representing ground-truth values of $\boldsymbol{\alpha}$'s), which were meant to emulate 1000 actual user-layouts or preferences. In each case, $\boldsymbol{\alpha}$-estimation was performed to recover the original values as best as possible. Note that this recovery is only partially effective due to the information loss in projecting down to a 2-D

[3] In addition to using random weights as a control we have also tested other default weighting schemes such as using *equal* weights, which have yielded similar benchmarking results. Nevertheless, we have used random weighting in order to minimize any systematic bias that might arise due to a deterministic weighting scheme.

space. As a control, 1000 randomly generated feature weights were used to see how well they could match the user layouts (i.e., by chance alone).

Our primary test database consists of 142 images from the COREL database. It has 7 categories of car, bird, tiger, mountain, flower, church and airplane. Each class has about 20 images. Feature extraction based on color, texture and structure has been done off-line and pre-stored. Although we will be reporting on this test data set -- due to its common use and familiarity to the CBIR community -- we should emphasize that we have also successfully tested our methodology on larger and much more heterogeneous image libraries. For example: real personal photo collections of 500+ images (including family, friends, vacations, etc.).

The following is the Monte Carlo procedure that was used for testing the significance and validity of user-modeling with $\boldsymbol{\alpha}$-estimation:

1. Randomly select *M* images from the database. Generate arbitrary (random) feature weights $\boldsymbol{\alpha}$ in order to simulate a “user” layout.
2. Do a PCA Splat using this “ground truth” $\boldsymbol{\alpha}$.
3. From the resulting 2-D layout, estimate $\boldsymbol{\alpha}$ and denoted the estimated $\boldsymbol{\alpha}$ as $\hat{\boldsymbol{\alpha}}$.
4. Select a new distinct (non-overlapping) set of *M* images from the database.
5. Do PCA Splats on the second set using the original $\boldsymbol{\alpha}$, the estimated $\hat{\boldsymbol{\alpha}}$ and a third random $\boldsymbol{\alpha}'$ (as control).
6. Calculate the resulting stress in Eq. (8), and layout deviation (2-D position error) in Eq. (9) for the original, estimated and random (control) values of $\boldsymbol{\alpha}$, $\hat{\boldsymbol{\alpha}}$, and $\boldsymbol{\alpha}'$, respectively.
7. Repeat 1000 times

The scatter-plot of $\boldsymbol{\alpha}$-estimation is shown in Figure 13. Clearly there is a direct linear relationship between the original weights $\boldsymbol{\alpha}$ and the estimated weights $\hat{\boldsymbol{\alpha}}$. Note that when the original weight is very small (<0.1) or very large (>0.9), the estimated weight is zero or one correspondingly. This means that when one particular feature weight is very large (or very small), the corresponding feature will become the most dominant (or least dominant) feature in the PCA, therefore the estimated weight for this feature will be either 1 or 0. This “saturation” phenomenon in Figure 13 is seen to occur more prominently for the case of structure (lower left of the rightmost panel) that is possibly more pronounced because of the structure feature vector being so (relatively) high-dimensional. Additionally, structure features are not as well defined compared with color and texture (e.g., they have less discriminating power).

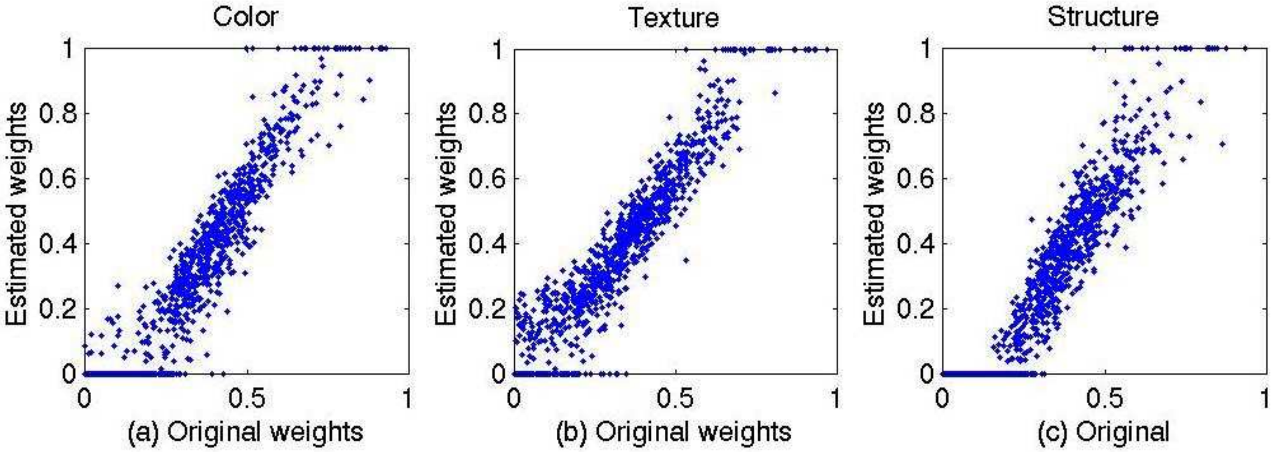


Figure 13. Scatter-plot of $\boldsymbol{\alpha}$-estimation: Estimated weights vs. original weights

In terms of actual measures of stress and deviation we found that the $\boldsymbol{\alpha}$-estimation scheme yielded the smaller deviation 78.4% of the time and smaller stress 72.9%. The main reason these values are less than 100% is due to the nature of the Monte Carlo testing and the fact that working with low-dimensional (2-D) spaces, random weights can be close to the original weights and hence can often generate similar "user" layouts (in this case apparently about 25% of the time).

We should add that an alternative "control" or null hypothesis to that of random weights is that of fixed equal weights. This "weighting" scheme corresponds to the assumption that there are to be no preferential biases in the subspace of the features, that they should all count equally in forming the final layout (or default PCA). In an identical set of experiments, replacing random weights for comparison layouts with equal weights $\boldsymbol{\alpha} = \{\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\}^T$, we found a similar distribution of similarity scores. In particular, since positional accuracy is the most direct measure of layout effectiveness, we noted that the $\boldsymbol{\alpha}$-estimation scheme yielded the smaller deviation 72.6% of the time compared to equal weights (as opposed to the 78.4% compared with random weights). We therefore note that the results and conclusions of these experiments are consistent despite the choice of equal or random controls and ultimately direct $\boldsymbol{\alpha}$-estimation of a "user" layout is best.

Another control other than random (or equal) weights is to compare the deviation of an $\boldsymbol{\alpha}$-estimation layout generator to a simple scheme which assigns each new image to the 2-D location of its (un-weighted or equally weighted) 37-dimensional nearest-neighbor from the set of images previously laid out by the "user". This control scheme essentially operates on the principle that new images should be positioned on screen at the same location as their nearest neighbors in the original 37-dimensional feature space (the default similarity measure in the absence of any prior bias) and thus essentially ignores the operating subspace defined by the "user" in a 2-D layout.

The distributions of the outcomes of this Monte Carlo simulation are shown in Figure 14 where we see that the layout deviation using α-estimation (red: mean=0.9691 std=0.7776) was consistently lower -- by almost an order of magnitude -- than the nearest neighbor layout approach (blue: mean=7.5921, std=2.6410). We note that despite the non-coincident overlap of the distributions' tails in Figure 14, in every one of the 1000 random trials the α-estimation deviation score was found to be smaller than that of nearest-neighbour (a key fact not visible in such a plot).

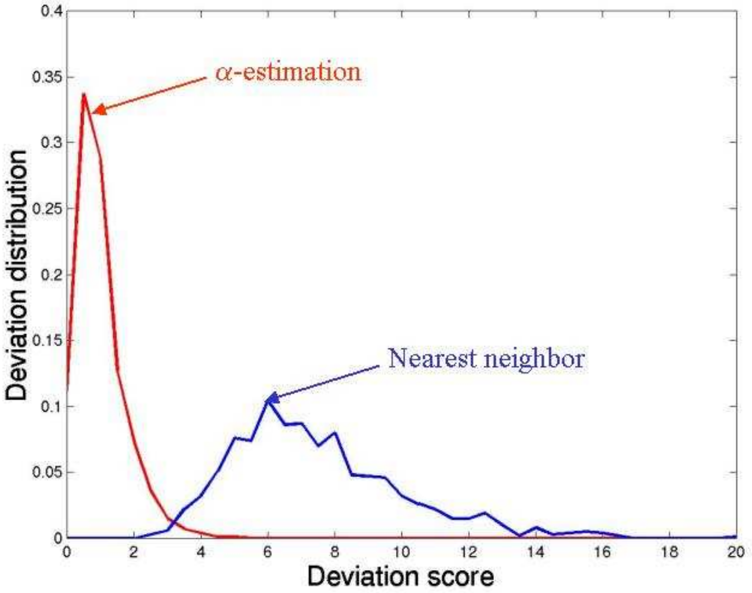


Figure 14. Comparison of the distribution of α-estimation vs. nearest neighbor deviation scores

5 User Preference Study

In addition to the computer-generated simulations, we have in fact conducted a preliminary user study that has also demonstrated the superior performance of α-estimation over random feature weighting used as a control. The goal was to test whether the estimated feature weights would generate a better layout on a *new* but similar set of images than random weightings (used as control). The user interface is shown in Figure 15 where the top panel is a coherent layout generated by a random α on reference image set. From this layout, an estimate of α was computed and used to redo the layout. A layout generated according to random weights was also generated and used as a control. These two layouts were then displayed in the bottom panels with randomized (A vs. B) labels (in order to remove any bias in the presentation). The user's task was to select which layout (A or B) was more similar to the reference layout in the top panel.

In our experiment, 6 naïve users were instructed in the basic operation of the interface and given the following instructions: (1) both absolute and relative positions of images matter, (2) in general, similar images, like cars, tigers, etc., should cluster and (3) the relative positions of the clusters also matter. Each user performed 50 forced-choice tests with no time limits. Each test set of 50 contained redundant (randomly reoccurring) tests in order to test the user's consistency. We specifically aimed at not "priming" the subjects with very detailed instructions (such as "it's not valid to match a red car and a red flower because they are both red"). In fact, the "naïve" test subjects were told nothing at all about the 3 feature types (color, texture, structure), the associated α or obviously the estimation technique. In this regard, the paucity of the instructions was entirely intentional: whatever mental grouping that seemed valid to them was the key. In fact, this very same flexible association of the user is what was specifically tested for in the *consistency* part of the study.

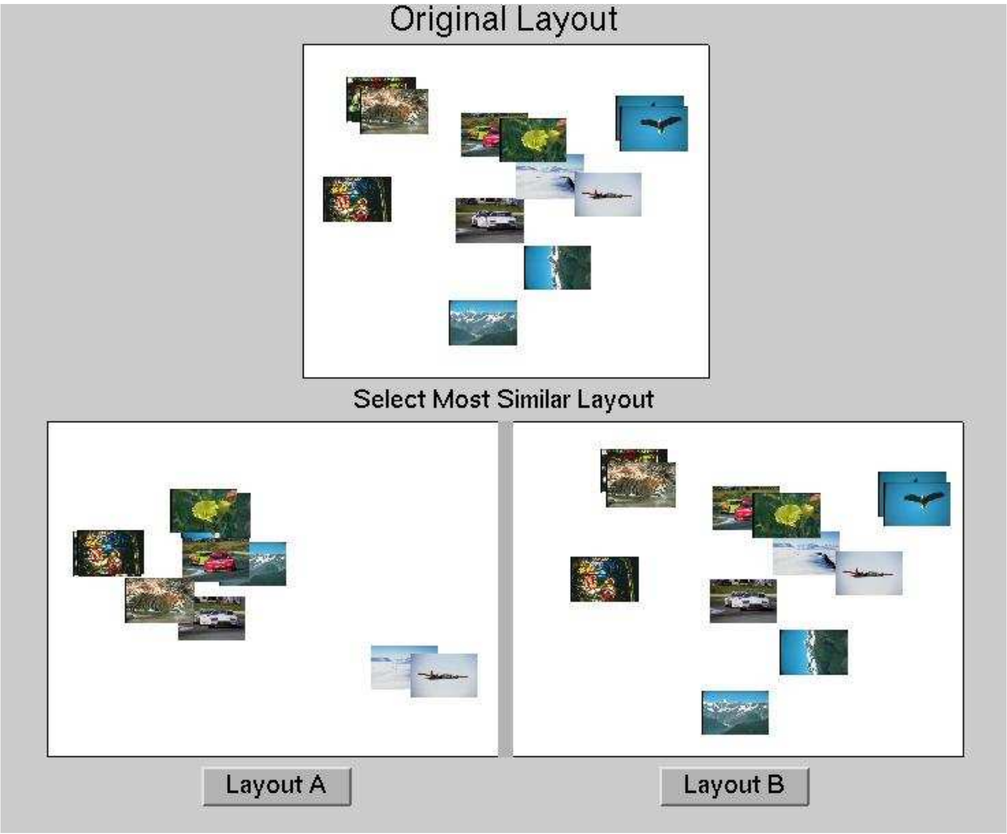


Figure 15. α -estimation-matters user test interface

Table 1 shows the results of this user-study. The average preference indicated for the $\boldsymbol{\alpha}$-estimation-based layout was found to be 96% and an average consistency rate of a user was 97%. We note that the $\boldsymbol{\alpha}$-estimation method of generating a layout in a similar "style" to the reference was consistently favored by the users. A similar experimental study has shown this to also be true even if the test layouts consist of *different* images than those used in the reference layout (i.e. similar but not identical images from the same categories or classes).

Table 1. Results of user-preference study

	Preference for estimates	Preference for random weights	User's consistency rate
User 1	90%	10%	100%
User 2	98%	2%	90%
User 3	98%	2%	90%
User 4	95%	5%	100%
User 5	98%	2%	100%
User 6	98%	2%	100%
Average	**96%**	**4%**	**97%**

6 Discussion

There are several areas of future work. First, more extensive user-layout studies are needed to replace Monte Carlo simulations. It is critical to have real users with different notions of content do layouts and to see if our system can model them. Moreover, we have already integrated hybrid visual/semantic feature weighting which requires further testing with human subjects. We have designed our system with general CBIR in mind but more specifically for personalized photo collections. The final visualization and retrieval interface can be displayed on a computer screen, large panel projection screens, or -- for example -- on embedded tabletop devices [3, 4], designed specifically for purposes of story-telling or multi-person collaborative exploration of large image libraries.

We note that although the focus of this paper has been on visual content analysis, the same framework for visualization and user-modeling would apply to other data entities such as audio files, video clips, documents, or web pages, etc. The main difference would be the choice of features used, their representations in high-dimensional spaces and the appropriate metrics.

7 Conclusion

In this paper, we proposed an optimized content-based visualization technique to generate a 2-D display of the retrieved images for content-based image retrieval. We believe that both the computational results and the pilot user study support our claims of a more perceptually intuitive and informative visualization engine that not only provides a better understanding of query retrievals but also aids in forming new queries.

The proposed content-based visualization method can be easily applied to project the images from high-dimensional feature space to a 3-D space for more advanced visualization and navigation. Features can be multi-modal, expressing individual visual features, e.g., color alone, audio features and semantic features, e.g., keywords, or any combination of the above. The proposed layout optimization technique is also a quite general approach and can be applied to avoid overlapping of any type of images, windows, frames and boxes. For the future work, relevance feedback can be achieved based on the visualization of the optimized PCA Splat. By grouping the relevant images together, a new user-modeling technique will be proposed.

The PDH project is at its initial stage. We have just begun our work in both the user interface design and photo visualization and layout algorithms. Many interesting questions still remain as our future research in the area of content-based information visualization and retrieval. The next task is to carry out an extended user-modeling study by having our system learn the feature weights from various sample layouts provided by the user. We have already developed a framework to incorporate visual features with semantic labels for both retrieval and layout. Incorporation of relevance feedback in our framework seems very intuitive and is currently being explored. Another challenging area is automatic “summarization” and display of large image collections. Since summarization is implicitly defined by user preference, α-estimation for user-modeling will play a key role in this and other high-level tasks where context is defined by the user.

<u>Appendix A</u>

Define the following terms:

$$V_{(ij)}^{c} = \sum_{k=1}^{L_c} | \mathbf{X}_{c\,(i)}^{\,(k)} - \mathbf{X}_{t\,(j)}^{\,(k)} |^{p} \tag{A.1}$$

$$V_{(ij)}^{t} = \sum_{k=1}^{L_t} | \mathbf{X}_{t\,(i)}^{\,(k)} - \mathbf{X}_{t\,(j)}^{\,(k)} |^{p} \tag{A.2}$$

$$V_{(ij)}^{s} = \sum_{k=1}^{L_s} | \mathbf{X}_{s\,(i)}^{\,(k)} - \mathbf{X}_{s\,(j)}^{\,(k)} |^{p} \tag{A.3}$$

and subsequently simplify Eq. (1) to:

$$J = \sum_{i=1}^{N} \sum_{j=1}^{N} (d_{ij}{}^{p} - \alpha_c{}^{p} V_{(ij)}^{c} - \alpha_t{}^{p} V_{(ij)}^{t} - \alpha_s{}^{p} V_{(ij)}^{s})^{2} \tag{A.4}$$

To minimize *J*, we take the partial derivatives of *J* relative to α_c^p , α_t^p , α_s^p and set them to zero, respectively.

$$\begin{aligned} \frac{\partial J}{\partial \alpha_c^p} &= 0 \\ \frac{\partial J}{\partial \alpha_t^p} &= 0 \\ \frac{\partial J}{\partial \alpha_s^p} &= 0 \end{aligned} \tag{A.5}$$

We thus have:

$$\sum_{i=1}^{N}\sum_{j=1}^{N} V_{(ij)}^{c}{}^{2} \cdot \alpha_c^p + \sum_{i=1}^{N}\sum_{j=1}^{N} V_{(ij)}^{c} V_{(ij)}^{t} \cdot \alpha_t^p + \sum_{i=1}^{N}\sum_{j=1}^{N} V_{(ij)}^{c} V_{(ij)}^{s} \cdot \alpha_s^p = \sum_{i=1}^{N}\sum_{j=1}^{N} d_{ij}{}^{2} V_{(ij)}^{c} \tag{A.6}$$

$$\sum_{i=1}^{N}\sum_{j=1}^{N} V_{(ij)}^{c} V_{(ij)}^{t} \cdot \alpha_c^p + \sum_{i=1}^{N}\sum_{j=1}^{N} V_{(ij)}^{t}{}^{2} \cdot \alpha_t^p + \sum_{i=1}^{N}\sum_{j=1}^{N} V_{(ij)}^{t} V_{(ij)}^{s} \cdot \alpha_s^p = \sum_{i=1}^{N}\sum_{j=1}^{N} d_{ij}{}^{2} V_{(ij)}^{t} \tag{A.7}$$

$$\sum_{i=1}^{N}\sum_{j=1}^{N} V_{(ij)}^{c} V_{(ij)}^{s} \cdot \alpha_c^p + \sum_{i=1}^{N}\sum_{j=1}^{N} V_{(ij)}^{t} V_{(ij)}^{s} \cdot \alpha_t^p + \sum_{i=1}^{N}\sum_{j=1}^{N} V_{(ij)}^{s}{}^{2} \cdot \alpha_s^p = \sum_{i=1}^{N}\sum_{j=1}^{N} d_{ij}{}^{2} V_{(ij)}^{s} \tag{A.8}$$

Using the following matrix/vector definitions

$$A = \begin{bmatrix} \sum_{i=1}^{N}\sum_{j=1}^{N} V_{(ij)}^{c\ 2} & \sum_{i=1}^{N}\sum_{j=1}^{N} V_{(ij)}^{c} V_{(ij)}^{t} & \sum_{i=1}^{N}\sum_{j=1}^{N} V_{(ij)}^{c} V_{(ij)}^{s} \\ \sum_{i=1}^{N}\sum_{j=1}^{N} V_{(ij)}^{c} V_{(ij)}^{t} & \sum_{i=1}^{N}\sum_{j=1}^{N} V_{(ij)}^{t\ 2} & \sum_{i=1}^{N}\sum_{j=1}^{N} V_{(ij)}^{t} V_{(ij)}^{s} \\ \sum_{i=1}^{N}\sum_{j=1}^{N} V_{(ij)}^{c} V_{(ij)}^{s} & \sum_{i=1}^{N}\sum_{j=1}^{N} V_{(ij)}^{t} V_{(ij)}^{s} & \sum_{i=1}^{N}\sum_{j=1}^{N} V_{(ij)}^{s\ 2} \end{bmatrix} \quad \boldsymbol{\beta} = \begin{bmatrix} \alpha_c^p \\ \alpha_t^p \\ \alpha_s^p \end{bmatrix} \text{ and } b = \begin{bmatrix} \sum_{i=1}^{N}\sum_{j=1}^{N} d_{ij}{}^2 V_{(ij)}^{c} \\ \sum_{i=1}^{N}\sum_{j=1}^{N} d_{ij}{}^2 V_{(ij)}^{t} \\ \sum_{i=1}^{N}\sum_{j=1}^{N} d_{ij}{}^2 V_{(ij)}^{s} \end{bmatrix}$$

Equations (A.6-A.8) are easily simplified

$$A \cdot \boldsymbol{\beta} = b \tag{A.9}$$

Subsequently $\boldsymbol{\beta}$ is obtained as a constrained ($\boldsymbol{\beta}$>0) linear least-squares solution of the above system. The weighting vector $\boldsymbol{\alpha}$ is then simply determined by the pth root of $\boldsymbol{\beta}$ where we typically use p = 2.

Appendix B

Let $\mathbf{P}_i = p^{(i)}(x, y) = \begin{pmatrix} x_i \\ y_i \end{pmatrix}$ and $\hat{\mathbf{P}}_i = \hat{p}^{(i)}(x, y) = \begin{pmatrix} \hat{x}_i \\ \hat{y}_i \end{pmatrix}$, and Eq. (9) is rewritten as

$$J = \sum_{i=1}^{N} \| \mathbf{P}_i - \hat{\mathbf{P}}_{\mathrm{i}} \|^2 \tag{B.1}$$

Let $\mathbf{X}_{\mathrm{i}}$ be the column feature vector of the ith image, where $\mathbf{X}_{\mathrm{i}} = \begin{bmatrix} \mathbf{X}_{\mathrm{c}}^{(\mathrm{i})} \\ \mathbf{X}_{\mathrm{t}}^{(\mathrm{i})} \\ \mathbf{X}_{\mathrm{s}}^{(\mathrm{i})} \end{bmatrix}$, $i = 1, \cdots, N$. $\mathbf{X}_{\mathrm{c}}^{(\mathrm{i})}$, $\mathbf{X}_{\mathrm{t}}^{(\mathrm{i})}$ and $\mathbf{X}_{\mathrm{s}}^{(\mathrm{i})}$ are the corresponding color, texture and structure feature vector of the ith image and their lengths are L_c, L_t and L_s, respectively. Let $\mathbf{X}'_{\mathrm{i}} = \begin{pmatrix} \alpha_{\mathbf{c}} \mathbf{X}_{\mathrm{c}}^{(\mathrm{i})} \\ \alpha_{\mathbf{t}} \mathbf{X}_{\mathrm{t}}^{(\mathrm{i})} \\ \alpha_{\mathrm{s}} \mathbf{X}_{\mathrm{s}}^{(\mathrm{i})} \end{pmatrix}$ be the weighted high-dimensional feature vector. These weights α_c, α_t, α_s are constrained such as they always sum to 1.

$\hat{\mathbf{P}}_i$ is estimated by linearly projecting the weighted high-dimensional features to 2-D dimension. Let $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_{\mathrm{N}}]$, it is a $L \times N$ matrix, where $L = L_c + L_t + L_s$. $\hat{\mathbf{P}}_i$ is estimated by

$$\hat{\mathbf{P}}_{\mathrm{i}} = U^T (\mathbf{X}'_{\mathrm{i}} - \mathbf{X}_{\mathrm{m}}) \qquad i = 1, \cdots, N \tag{B.2}$$

where U is a $L \times 2$ projection matrix, $\mathbf{X}_{\mathrm{m}}$ is a $L \times 1$ mean column vector of $\mathbf{X}'_{\mathbf{i}}$, $i = 1, \cdots, N$. Substitute $\hat{\mathbf{P}}_i$ by Eq (B.2) into Eq. (B.1), the problem is therefore one of seeking the optimal feature weights $\boldsymbol{\alpha}$, projection matrix *U*, and column vector $\mathbf{X}_m$ such as *J* in Eq. (B.3) is minimized, given $\mathbf{X}_{\mathrm{i}}$, $\mathbf{P}_{\mathrm{i}}$, $i = 1, \cdots, N$.

$$J = \sum_{i=1}^{N} \| U^T (\mathbf{X}'_{\mathrm{i}} - \mathbf{X}_m) - \mathbf{P}_{\mathrm{i}} \|^2 \tag{B.3}$$

In practice, it is almost impossible to estimate optimal $\boldsymbol{\alpha}$, *U* and $\mathbf{X}_m$ simultaneously based on the limited available data $\mathbf{X}_{\mathrm{i}}$, $\mathbf{P}_{\mathrm{i}}$, $i = 1, \cdots, N$. We thus make some modifications. Instead of estimating $\boldsymbol{\alpha}$, *U* and $\mathbf{X}_m$ simultaneously, we modified the estimation process to be a two-step *re-estimation* procedure. We first estimate the projection matrix *U* and column vector $\mathbf{X}_m$, and then estimate feature weight vector $\boldsymbol{\alpha}$ based on the computed *U* and $\mathbf{X}_m$, and iterate until convergence.

Let $U^{(0)}$ be the eigenvectors corresponding to the largest two eigenvalues of the covariance matrix of **X**, where $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_{\mathrm{N}}]$, $\mathbf{X}_{\mathrm{m}}^{(0)}$ be the mean vector of **X.**

We have

$$\mathbf{P}_{\mathrm{i}}^{(0)} = (U^{(0)})^T (\mathbf{X}_{\mathrm{i}} - \mathbf{X}_{\mathrm{m}}^{(0)}) \tag{B. 4}$$

$\mathbf{P}_{\mathrm{i}}^{(0)}$ is the projected 2D coordinates of the unweighted high dimensional feature vector of the *i*th image. Ideally its target location is $\mathbf{P}_{\mathrm{i}}$. To consider the alignment correction, a rigid transform [28] is applied.

$$\hat{\mathbf{P}}_{\mathrm{i}}^{(0)} = A \cdot \mathbf{P}_{\mathrm{i}}^{(0)} + T \tag{B. 5}$$

where *A* is a 2×2 matrix and *T* is a 2×1 vector. *A*, *T* are obtained by minimizing the L_2 -norm of $\mathbf{P}_{\mathrm{i}} - \hat{\mathbf{P}}_{\mathrm{i}}^{(0)}$.

Therefore *J* in (B. 3) is modified to

$$J = \sum_{i=1}^{N} \| AU^{(0)} (\mathbf{X}'_i - \mathbf{X}_{\mathrm{m}}^{(0)}) - (\mathbf{P}_{\mathrm{i}} - T) \|^2 \tag{B. 6}$$

Let $U = AU^{(0)}$, $\mathbf{X}_m = \mathbf{X}_\mathrm{m}^{(0)}$ and $\mathbf{P}_\mathrm{i} = \mathbf{P}_\mathrm{i} - T$, we still have the form of Eq. (B. 3).

Let us rewrite $U^T = \begin{bmatrix} U_{11}, \cdots U_{1(L_c+L_t+L_s)} \\ U_{21}, \cdots, U_{2(L_c+L_t+L_s)} \end{bmatrix}$

After some simplifications on Eq. (B. 3), we have

$$J = \sum_{i=1}^{N} \| \alpha_c A_i + \alpha_t B_i + \alpha_s C_i - D_i \|^2 \qquad \text{(B. 7)}$$

where $A_i = \begin{bmatrix} \sum_{k=1}^{L_c} U_{1k} \mathbf{X}_c{}^{(k)}(i) \\ \sum_{k=1}^{L_c} U_{2k} \mathbf{X}_c{}^{(k)}(i) \end{bmatrix}$ $B_i = \begin{bmatrix} \sum_{k=1}^{L_t} U_{1(k+L_c)} \mathbf{X}_t{}^{(k)}(i) \\ \sum_{k=1}^{L_t} U_{2(k+L_t)} \mathbf{X}_t{}^{(k)}(i) \end{bmatrix}$ $C_i = \begin{bmatrix} \sum_{k=1}^{L_s} U_{1(k+L_c+L_t)} \mathbf{X}_s{}^{(k)}(i) \\ \sum_{k=1}^{L_s} U_{2(k+L_c+L_t)} \mathbf{X}_s{}^{(k)}(i) \end{bmatrix}$ and

$D_i = U^T \mathbf{X}_m + \mathbf{P}_i$,

A_i, B_i, C_i and D_i are the 2×1 feature vectors, respectively.

To minimize *J*, we take the partial derivatives of *J* relative to α_c, α_t, α_s and set them to zero, respectively.

$$\begin{aligned} \frac{\partial J}{\partial \alpha_c} &= 0 \\ \frac{\partial J}{\partial \alpha_t} &= 0 \\ \frac{\partial J}{\partial \alpha_s} &= 0 \end{aligned} \qquad \text{(B. 8)}$$

We thus have:

$$E \cdot \boldsymbol{\alpha} = f \qquad \text{(B. 9)}$$

where $E = \begin{bmatrix} \sum_{i=1}^{N} A_i{}^T A_i & \sum_{i=1}^{N} A_i{}^T B_i & \sum_{i=1}^{N} A_i{}^T C_i \\ \sum_{i=1}^{N} B_i{}^T A_i & \sum_{i=1}^{N} B_i{}^T B_i & \sum_{i=1}^{N} B_i{}^T C_i \\ \sum_{i=1}^{N} C_i{}^T A_i & \sum_{i=1}^{N} C_i{}^T B_i & \sum_{i=1}^{N} C_i{}^T C_i \end{bmatrix}$, $f = \begin{bmatrix} \sum_{i=1}^{N} A_i{}^T D_i \\ \sum_{i=1}^{N} B_i{}^T D_i \\ \sum_{i=1}^{N} C_i{}^T D_i \end{bmatrix}$

and $\boldsymbol{\alpha}$ is obtained by solving the linear equations (B.9).

Acknowledgements

The authors would like to thank the anonymous reviewers. This work was supported in part by Mitsubishi Electric Research Laboratories (MERL) and the National Science Foundation Grants DA 96-24396 and EIA 99-75019.

References

[1] M. Balabanovic, L. Chu, and G. Wolff, "Storytelling with Digital Photographs," *Proceedings of CHI2000*, April 2000, The Hague, The Netherlands.

[2] P. Dietz and D. Leigh, "DiamondTouch: A Multi-User Touch Technology," *Proceedings of ACM UIST'01*, November 2001, Orlando, FL.

[3] C. Shen, N. Lesh, B. Moghaddam, P. Beardsley, and R. Bardsley, "Personal Digital Historian: User Interface Design," *Proceedings of Extended Abstract of CHI 2001*. pp. 29-30, April 2001, Seattle WA.

[4] C. Shen, N. Lesh, and F. Vernier, "Personal Digital Historian: Story Sharing Around the Table," *ACM Interactions*, March/April 2003 (also MERL TR2003-04).

[5] H. Kang and B. Shneiderman, "Visualization Methods for Personal Photo Collections: Browsing and Searching in the PhotoFinder," *Proceedings of IEEE International Conference on Multimedia and Expo* (ICME2000), New York City, New York

[6] F. Vernier, N. Lesh, and C. Shen, "Visualization Techniques for Circular Tabletop Interface," *Proceedings of AVI'02*, pp. 257-266, May 2002, Trento, Italy.

[7] Mimio, www.mimio.com/meet/mimiomouse

[8] C. Shen, N. Lesh, F. Vernier, C. Forlines, and J. Frost, "Sharing and Building Digital Group Histories," *ACM Conference on Computer Supported Cooperative Work (CSCW)*, November 2002.

[9] Q. Tian, B. Moghaddam, and T. S. Huang, "Display Optimization for Image Browsing," *2nd International Workshop on Multimedia Databases and Image Communications (MDIC'01)*, pp. 167-173, Sep. 17-18, 2001, Amalfi, Italy.

[10] B. Moghaddam, Q. Tian, and T. S. Huang, "Spatial Visualization for Content-Based Image Retrieval," *Proceedings of IEEE International Conference on Multimedia and Expo (ICME01)*, August 22-25, Tokyo, Japan, 2001.

[11] B. Moghaddam, Q. Tian, N. Lesh, C. Shen, and T. S. Huang, "PDH: A human-Centric Interface for Image Libraries," *Proceedings of IEEE International Conference on Multimedia and Expo (ICME02)*, vol. 1, pp. 901-904, August 2002.

[12] Q. Tian, B. Moghaddam, and T. S. Huang, "Visualization, Estimation and User-Modeling for Interactive Browsing of Image Libraries," *International Conference on Image and Video Retrieval (CIVR'02)*, pp. 7-16, London, UK, July 18-19, 2002.

[13] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, “Content-Based Image Retrieval at the End of the Early Years,” *IEEE Transactions on. Pattern Analysis and Machine Intelligence,* vol. 22, no. 12, Dec. 2000.

[14] A. Horoike and Y. Musha, “Similarity-Based Image Retrieval System with 3-D Visualization,” *Proceedings of IEEE Int’l Conf. on Multimedia and Expo,* vol. 2, pp. 769-772, New York, 2000.

[15] M. Nakazato and T. S. Huang, “3D MARS: Immersive Virtual Reality for Content-Based Image Retrieval,” *Proceedings of IEEE Int’l Conf. on Multimedia and Expo*, Tokyo, Japan, Aug. 22-25, 2001.

[16] S. Santini and Ramesh Jain, “Integrated Browsing and Querying for Image Databases,” *July-September Issue, IEEE Multimedia Magazine,* pp.26-39, 2000.

[17] S. Santini, A. Gupta, and R. Jain, “Emergent Semantics Through Interaction in Image Databases,” *IEEE Transactions on Knowledge and Data Engineering,* vol. 13, no. 3, pp. 337-351, May 2001.

[18] Y. Rubner, “Perceptual Metrics for Image Database Navigation,” Ph.D. dissertation, Stanford University, 1999.

[19] M. Stricker and M. Orengo, “Similarity of Color Images,” *Proc. SPIE Storage and Retrieval for Image and Video Databases,* 1995.

[20] 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.

[21] S. X. Zhou, Y. Rui, and T. S. Huang, “Water-filling Algorithm: A Novel Way for Image Feature Extraction Based on Edge Maps,” *Proc. IEEE Int’l. Conf. On Image Proc.*, Japan, 1999.

[22] S. Santini and R. Jain, “Similarity Measures,” *IEEE Transactions on PAMI,* vol. 21, no. 9, 1999.

[23] M. Popescu and P. Gader, “Image Content Retrieval From Image Databases Using Feature Integration by Choquet Integral,” *SPIE Conference Storage and Retrieval for Image and Video Databases VII*, San Jose, CA, 1998.

[24] D. M. Squire, H. Müller, and W. Müller, “Improving Response Time by Search Pruning in a Content-Based Image Retrieval System, Using Inverted File Techniques”, *Proc. of IEEE workshop on CBAIVL,* June 1999.

[25] D. Swets and J. Weng, “Hierarchical Discriminant Analysis for Image Retrieval,” *IEEE Transactions on PAMI*, vol. 21, no.5, 1999.

[26] W. S. Torgeson, *Theory and methods of scaling*, John Wiley and Sons, New York, NY, 1958.

[27] I. T. Jolliffe, *Principal Component Analysis*, Springer-Verlag, New-York, 1986.

[28] D. Zwillinger (Ed.), “Affine Transformations”, §4.3.2 in “CRC Standard Mathematical Tables and Formulae”, Boca Raton, FL: CRC Press, pp. 265-266, 1995.