Organizing and Browsing Photos using Different Feature Vectors and Their Evaluations

Grant Strong
Memorial University of Newfoundland
Dept. of Computer Science, MUN
St. John's, NL, Canada
+1 709 737-6947
strong@cs.mun.ca

Minglun Gong
Memorial University of Newfoundland
Dept. of Computer Science, MUN
St. John's, NL, Canada
+1 709 737-3589
gong@cs.mun.ca

ABSTRACT
Two-dimensional similarity-based image organizing studies how to place photos within 2D virtual canvas based on their visual contents so that the users can easily locate the desired photos. As an extension to our previous work [10], several improvements are made in this paper to allow better photo browsing experiences. For example, the new approach pre-orders all the photos so that a consistent set of photos is selected for display. This solves the photo flickering problem of our previous approach, which uses K-mean algorithm to dynamically select photos.

The main focus of this paper however is on the evaluation of the effectiveness of different feature vectors for 2D photo organization. A performance metric is proposed to measure how well photos with similar visual contents are grouped together on the 2D canvas. Feature vectors generated using eight different low-level feature extraction approaches are tested. The evaluation results reveal the pros and cons of different feature extraction approaches, which can be a useful guide for developing new feature vectors.

Categories and Subject Descriptors
I.4.10 [IMAGE PROCESSING AND COMPUTER VISION]: Image Representation – m*ultidimensional, statistical*

General Terms
Algorithms, Performance

Keywords
Similarity-based photo browsing, 2D similarity-based photo organizing, self organizing map, performance evaluation

1. INTRODUCTION
With the increasing easiness of capturing digital photographs, personal photo collections have been getting larger and larger. It is not unusual for a regular person having tens of thousands photos in his/her computer. Unfortunately, the techniques for photo organizing and browsing have not advanced in a matching speed. Photos are stored in the computer the same way as other types of files. When looking for a particular photo without knowing its file name, users need to first find the right folder and then browse through all the photos in the folder to find the right one.

Several techniques [3, 9, 12] have been proposed to facilitate active exploration of a collection of photos by organizing them based on their visual similarity. The paper presented here is an extension to our previous work in this area. In our previous work [10], we proposed to organize photos on 2D virtual canvas according to their similarities. Photo content is first analyzed and then transformed into a feature vector for arrangement processing. A Self Organizing Map (SOM) is then used to find the optimal location for each photo so that the ones with similar feature vectors are closer to each other. To speed up the SOM training processing, the algorithm is implemented on the Graphics Processing Unit (GPU) of programmable graphics hardware. In the end, the users can interactively organize thousands of photos and browse them through intuitive operations, such as pan and zoom [10].

While the SOM can effectively map high dimensional space to 2D canvas in a topology preserving way, it relies on the feature extraction approach to assign similar feature vectors to visually similar photos. The final organization can only be as good as how well the feature vectors model the photo's similarity to one another. Hence, which feature extraction approach is best suited for similarity-based photo organization is worth investigating.

Until now the normal way of evaluating feature extracting approaches was through statistic measures, such as precision and recall. These measures are suitable for image retrieval applications, where users know what type of photos they want. However when the goal is for image browsing, these measures do not offer an intuitive evaluation on how well visually similar photos are grouped together. In this paper, we propose a new measure of feature vector effectiveness that quantifies similarity-based photo organizing potential. This photo browsing index differs from other common measures as it does not focus on retrieval accuracy but rather the aesthetic appeal of the organization that can be achieved with a feature vector type from the user's perspective. When browsing a collection of photos users prefer that photos of similar content category be clustered in the same relative region during visualization. Springing from this idea, our evaluation metric is derived based on the average distance of photos in each category.

CIVR '09, July 8-10, 2009 Santorini, GR

2. RELATED WORK

2.1 Content-based Image Retrieval

Content based image retrieval (CBIR) has been an active research area in the past decade [8]. Based on the user interaction required, existing approaches can be classified into two categories: query by example (QBE) and relevance feedback (RF). QBE approaches require the user to present an example image (or a hand-drawn sketch) as a visual query [2], whereas the RF approaches require the user to provide feedback (relevant or irrelevant) information on multiple images [13], which is used to model what the user is looking for.

The proposed image organizing and browsing algorithm is similar to existing QBE approaches since in both cases the performance depends on the low-level features extracted from images. Many advanced feature extraction techniques have been proposed and their relative performances for CBIR are evaluated [11]. However we argue that the precision/recall based evaluation results may not help for similarity-based photo organizing application, which has different objectives. In addition, these advanced feature extraction techniques are generally computationally expensive. When the goal is to interactively organize a dynamic set of photos, such as the ones returned by online image search engine, simple feature vectors have to be used so that hundreds of photos can be analyzed at interactive speed.

2.2 Similarity-based Image Browsing

While CBIR approaches are for users with clear goal about what they are searching for, similarity-based image browsing approaches aim for users who just want to surf/browse an image collection [3]. The challenge here is how to arrange images based on visual similarities.

Several approaches have been proposed for browsing images. Torres et al. use both spiral-based and concentric-based representations to display images based on similarities, with more similar ones closer to the center [12]. Chen et al. visualize image database using pathfinder networks in which similar images are linked together. Snavely et al arrange community photos based on the viewpoints used for capturing the photos [9]. Our previous work [10] falls into this category and is used as a tool for the feature vector evaluation.

2.3 Self Organizing Map

A self organizing map (SOM) is a special type of artificial neural network which consists of a set of interconnected units [5]. Once trained using unsupervised learning, a SOM can represent a set of high-dimensional samples in a topology preserving way, meaning similar samples will be placed together whereas dissimilar ones will be pushed away from each other. As a result, the SOM is useful for clustering and visualizing high-dimensional data, such as the feature vectors of photos.

SOM-based algorithms have been proposed for both image retrieval [6] and image browsing [7]. Both approaches use tree-structured SOM to organize images. Due to the high computation cost for training, the SOMs used in these two approaches are quite small in size. Same as our previous approach [10], here the GPU is used to train large SOMs at interactive speed. The large SOM size ensures that photos with distinct features are mapped unique units in the SOM. The coordinates of these units are used for evaluate how close visually similar photos are grouped together.

3. SIMILARITY-BASED PHOTO ORGANAZING AND BROWSING

The objective of similarity-base photo organizing is to position photos on a 2D virtual canvas based on their visually similarities. We argue that solving 2D photo organizing problem has important and practical applications. For example, operating systems can adopt this technique as an option to arrange photos in a folder by their visual contents, rather than the visually meaningless file names or sizes. Online photo sharing sites and image search engines can organize images by visual similarities so that the users can easily locate the images they have in mind.

To facilitate the proposing of new evaluation metrics, here we formally defined the similarity-based photo organizing problem as:

Given a set of photos, T, assign a 2D coordinate (x_j, y_j) to each photo $j, (j \in T)$, so that 1) the distance between two visually similar photos are as small as possible; and 2) overall photos are evenly distributed to make full use of display space.

3.1 Photo Organizing and Browsing Tool

In order to assign 2D coordinates to photos based on visual content, we first apply a feature extraction algorithm so that the visual appearance of a photo can be represented using a feature vector. The choices of feature extraction algorithms and their performances are the focus of this paper and will be discussed in Section 4. Here we explain how to organize and browse photos based on extracted feature vectors.

Once the high dimensional feature vectors for all photos are generated, they are mapped onto the 2D virtual canvas using the GPU-based SOM training algorithm that we previously proposed. The mapping is topology preserving so that feature vectors closer in the feature space are also closer on the 2D canvas. Interested readers please refer to [10] for the details of the SOM training algorithm.

The SOM training algorithm assigns a 2D coordinate (x_p, y_p) to each photo p, based on which the browsing tool displays the photo. When the number of photos in the collection is high, it is infeasible to display all photos to the scene at a reasonable resolution. To address this problem, a photo collage is dynamically generated using selected photos. The browsing interface allows users to use intuitive operations, such as pan and zoom, to locate photos of interest. The photo collage is automatically updated as the user moves around.

Here, an important improvement over our previous approach is how the photo collage is generated. In [10], the photo collage is generated based on the K-mean clustering algorithm. That is, when there are M photos within the current viewport and only N photos can be displayed, the K-mean algorithm is used to classify the photos into N clusters and the photos at the center of each cluster are shown. While this approach works well in general, it cannot guarantee the consistency of photos that are selected for display when the user slightly zooms or pans the 2D canvas, resulting photos flickering on and off.

To solve this problem, an extra step is performed to pre-order photos, i.e., to assign a static order to all photos in the collection so that when only N photos can be displayed, the first N visible photos in the order will be used to compose the photo collage. The ordering is based on the following two criteria:

- The photos that are more representative should have higher priorities to allow them to be selected first.
- Photos with similar priorities should be ordered in a dispersive way so that the photos selected for display are spread across the screen evenly.

The priorities of different photos are determined using a multi-resolution SOM. The bottom SOM, i.e. the one with the highest resolution, is obtained using the above described SOM training procedure. The upper level SOMs are generated from lower level SOMs directly without training. This is done by assigning each unit in an upper level SOM a weight vector that equals to the average of its children's weight vectors in the lower level SOM. The average weight vector is then used to find a best matching photo for each unit in the upper level SOMs. Since these photos have feature vectors that are closer to the average feature vector of its neighborhood, we consider these photos more representative and hence give them higher priority for display. Here we simply assign each photo j a priority value z_j that equals to the highest level in the multi-resolution SOM that the photo is mapped to.

To ensure the subset of photos selected spreads uniformly across the screen, the photos are also disperse ordered. Here we borrow the idea from image halftoning and pre-calculate a dispersed-dot dithering matrix of the same dimension as the bottom level SOM. When photo j is mapped to unit (x_j, y_j) in the bottom level SOM, it will be assigned a disperse order d_j, which equals to the value at (x_j, y_j) of the dithering matrix. The property of dispersed-dot dithering matrix ensures that when the first N photos selected for display spreads uniformly across the SOM.

After the above two steps, each photo j in the collection is assigned with a 4D vector (x_j, y_j, z_j, d_j). When selecting among visible photos to composite photo collage, the ones with smaller z values are selected first. Then for photos with the same z value, the ones with smaller d values are selected first. Once selected for display, the photo is texture mapped to a rectangle centered at location (x_j, y_j).

3.2 The Proposed Evaluation Metrics

To allow quantitative evaluation we assume that a ground truth classification of photos is available. That is, the photo collection T is manually divided into multiple subsets S_k, $\bigcup_{1\le k\le N} S_k = T$, so that photos within each subset are considered as visually similar. The quality of photo organizing result can then be evaluated using the span of the area occupied by photos in each subset.

To evaluate a given subset k, we first need to find the centroid photo for this subset. That is, find a photo $c_k, (c_k \in S_k)$ that satisfies the following condition:

$$\forall j (j \in S_k \wedge j \neq c_k), \sum_{i\in S_k} D(i, c_k) < \sum_{i\in S_k} D(i, j)$$

where $D(i, j)$ is the Euclidean distance between the coordinates of photos i and j.

Then, the span of the area occupied by photos in subset k is measured using the average distance between each photo in subset k and the centroid:

$$R_k(S_k) = \frac{1}{\|S_k\|} \sum_{i\in S_k} D(i, c_k)$$

where $\|S_k\|$ represents the number of photos in subset S_k. This metric tells us how close photos in a given subset are. It is worth noting that the average distance, instead of the maximum distance, is used because it is less prone to outliners.

On the other hand, we can calculate the average distance for all photos not in the subset k using:

$$R_k(T - S_k) = \frac{1}{\|T - S_k\|} \sum_{i\notin S_k} D(i, c_k)$$

where $\|T - S_k\|$ represents the number of photos not in the subset S_k. This metric tells us how far away that photos not belonging to a given subset are.

The ratio of the above two measures is a dimensionless number, which is a good evaluator of the effectiveness of the photo organization for subset S_k:

$$E_k = \frac{R_k(T - S_k)}{R_k(S_k)}$$

where E_k is referred as the effectiveness of the feature vector being used on photo subset S_k. An average effectiveness measure across the entire set of photos is then simply:

$$E = \frac{1}{\|N\|} \sum_{k=1}^{N} E_k$$

where the higher the E value the better the overall organization is, as it indicates that the photos from the same subset are close to each other whereas the photos not belonging to the same subset are far away.

The above effectiveness measure evaluates how well the first objective for photo organizing is satisfied. A similar approach can also be used for evaluate the second objects, i.e., whether photos from all of the different subsets are distributed across the whole canvas. That is, we can calculate the average span of all photos in the collection using:

$$F = \frac{1}{\|T\|} \sum_{i\in T} D(i, c)$$

where $\|T\|$ is the total number of photos and c is the centroid photo for the whole set T. Since the 2D virtual canvas size we used is normalized to 2×2, evenly distributed photos will give a F measure about 0.75 in value.

4. FEATURE VECTORS UNDER STUDY

This study focuses on the use of features vectors in the application of photo organizing and browsing. For this reason the feature vectors discussed are chosen to be as robust to superficial image modifications like translation and scaling as possible, as well as their being fast to compute since any system that is doing the organizing will be accessed by users with finite patience that expect reasonably interactive processing speeds.

The notation used in the following sections will be as follows:

- I denotes an image or more formally a set of pixels p that each have a color (r_p, g_p, b_p).

- $C(p) = \lfloor N_R \times r_p/256 \rfloor \times N_G \times N_B + \lfloor N_G \times g_p/256 \rfloor \times N_B + \lfloor N_B \times b_p/256 \rfloor$ is the quantized color of pixel p, where N_R, N_G, N_B are the total number of bins used for quantizing the RGB color channels, respectively.
- $\|S\|$ denotes cardinality of a set S.
- $\vec{p}$ denotes the 2D coordinates of pixel p in I.
- $|\vec{a} - \vec{b}|$ denotes the Manhattan distance between two vectors $\vec{a}$ and $\vec{b}$.
- $\Omega^h(p) = \{q | q \in I \wedge q \neq p \wedge |\vec{p} - \vec{q}| \leq h\}$ is the set of pixels within the $h \times h$ neighborhood of pixel p.
- G is the gradient map of image I and $G(p) = (\theta_p, l_p)$ keeps the gradient direction and magnitude for pixel p.
- $\Theta(p) = \lfloor N_\Theta \times \theta_p/2\pi \rfloor$ is the quantized gradient direction for pixel p, where N_Θ is the number of bins used for quantizing directions.
- $L(p) = \lfloor N_L \times l_p/L_{\max} \rfloor$ is the quantized gradient magnitude, where $L_{\max}$ is the maximum possible gradient magnitude and N_L is the number of bins used for quantizing directions.

4.1 Color-based Approaches

Color-based features are the dominating ones when images are small and shape is unperceivable or are displayed in a large group and details are less focused upon.

4.1.1 Color Histogram

The baseline of all image feature vectors is the color histogram. Each dimension of the final vector is the relative number of times that a certain color occurs in the image. By simply summing up the occurrences of colors in the image and normalizing these sums by the total number of pixels we have a feature vector that is robust to rotational and scaling transformations. The main caveat when using the color histogram is caused by the same property that makes it robust — its lack of spatial information. An example of this issue is that you can randomly shuffle the pixels of an image and get the same histogram, which is undesirable. The N-dimensional color histogram feature vector $\vec{v}$ can be calculated as follows:

$$\vec{v} = \frac{1}{\|I\|}[n_1, \cdots, n_N]$$
$$n_c = \|\{p | p \in I \wedge C(p) = c\}\|$$

where the vector's dimension is decided by $N = N_R \times N_G \times N_B$.

4.1.2 Color Autocorrelogram

The color autocorrelogram [4] extends the color histogram by introducing spatial information in the form of neighboring color probability. Each dimension of the final vector is the probability of finding two neighboring pixels of a certain color in the image. It is computed by counting the number of pairs of neighboring pixels that have the same color and then dividing by the total number of pairs that were checked. Since the spatial information in this type of vector is relative, it is robust to rotational and scaling transformations. The problem with this type of vector is it does not account for shape in the image.

The original color autocorrelogram considers multiple neighborhood size samples [4]. Our experiments show that a single neighborhood size sample is often more effective and hence the autocorrelogram calculation is simplified to:

$$\vec{v} = \left[\frac{n_1}{m_1}, \cdots, \frac{n_N}{m_N}\right]$$
$$m_c = \|\{(p,q) | C(p) = c \wedge q \in \Omega^2(p)\}\|$$
$$n_c = \|\{(p,q) | C(p) = C(q) = c \wedge q \in \Omega^2(p)\}\|$$

where the dimension of the vector is also decided by $N = N_R \times N_G \times N_B$.

4.2 Gradient-based Approaches

Gradient based feature vectors organize by shape, which is useful, but when considered in solitary can potentially lead to unobvious results when an organization is viewed from afar and the shape in photos is less apparent.

In all cases the initial gradient information is extracted from a grayscale version of the image by first convolving it with Sobel filters. The horizontal and vertical gradient values are then converted into direction (angle) and magnitude for every pixel in the image. Since gradient direction is linked to image rotation, the resulting feature vectors are not rotationally invariant. Feature vector of a rotated image contains the same set of values in the same order but these values will be translated in the vector with respect to the amount of rotation applied to the image [1]. Vectors based on gradients are however robust to scaling until the resolution of the image becomes too low to display the original shape, for example a circle becomes a dot.

4.2.1 Gradient Histogram

This is a direct adaptation of the color histogram to gradient data derived from an image. The directions and magnitudes of possible gradients are quantized into a number of bins. Each dimension in the feature vector represents the number of occurrences of gradient values belonging to the corresponding bin. Similar to color histogram, the occurrences are normalized based on the total number of samples available:

$$\vec{v} = \frac{1}{\|G\|}[n_1, \cdots, n_N]$$
$$n_i = \|\{p | L(p) \times N_\Theta + \Theta(p) = i\}\|$$

where the dimension of the vector is decided by $N = N_\Theta \times N_L$.

4.2.2 Gradient Autocorrelogram

This vector extends the probabilistic nature of the color autocorrelogram to the gradients. The gradient information is first quantized, as seen in the previous case, and then the neighboring occurrences of gradients are counted, summed into a final vector position, and divided by the total pairs of neighbors checked. The feature vector extracted measures how an image's gradient changes within the neighborhood.

$$\vec{v} = \left[\frac{n_1}{m_1}, \cdots, \frac{n_N}{m_N}\right]$$
$$m_i = \|\{(p,q) | L(p) \times N_\Theta + \Theta(p) = i \wedge q \in \Omega^2(p)\}\|$$
$$n_i = \left\|\left\{(p,q) \middle| \begin{array}{c} L(p) = L(q) \wedge \Theta(p) = \Theta(q) \wedge \\ L(p) \times N_\Theta + \Theta(p) = i \wedge q \in \Omega^2(p) \end{array}\right\}\right\|$$

where the vector's dimension is also decided by $N = N_\Theta \times N_L$.

4.2.3 Gradient Direction Histogram

Like the standard gradient histogram, this version models the global distribution of the gradient vectors but where it differs it eliminates the quantization of the gradient magnitude. Each dimension of the final vector represents the sum of all gradient magnitudes along a certain direction. The final values are normalized by the sum of all of the magnitudes. When the total feature vector dimension is fixed, this histogram allows the gradient direction to be quantized in finer resolution than the gradient histogram does. The computation of gradient direction histogram can be represented as:

$$\vec{v} = \frac{[m_1, \cdots, m_N]}{\sum_{i=1}^{N} m_i}$$
$$m_i = \sum_{p \in G, \Theta(p)=i} l_p$$

where the dimension of the vector $N = N_\Theta$.

4.2.4 *Gradient Direction Autocorrelogram*

Similar to gradient direction histogram, an autocorrelogram can also be defined using the gradient direction only, without considering the gradient magnitude. The corresponding feature vectors depict how edges in a given image change orientations within the local windows.

$$\vec{v} = \left[\frac{n_1}{m_1}, \cdots, \frac{n_N}{m_N}\right]$$
$$m_i = \|\{(p,q) | \Theta(p) = i \wedge q \in \Omega^2(p)\}\|$$
$$n_i = \|\{(p,q) | \Theta(p) = \Theta(q) = i \wedge q \in \Omega^2(p)\}\|$$

where we also have $N = N_\Theta$.

4.3 Hybrid Approaches

The visual content of an image is likely described by both color and gradient information. Hence, feature vectors based on either color or gradient only may not provide the best photo organizing results. Here two hybrid approaches are proposed which combines color- and gradient-based approaches in two different ways.

4.3.1 *Color Histogram + Gradient Direction Autocorrelogram Aggregation*

In this approach, the feature vector is split into two portions. The first potion is generated using the color histogram and the second is obtained using gradient direction autocorrelogram. Our experiments, discussed in the next section, suggest that, to achieve the optimal performance, the gradient-based approach generally requires feature vectors with higher dimensions than the color-based approach does. Hence here we allocate roughly 1/4 of the feature vector to store the color histogram information and 3/4 for the gradient direction autocorrelogram information. That is: $N = N_R \times N_G \times N_B + N_\Theta$ and $N_\Theta \approx {}^{3}/_{4} N$.

4.3.2 *Color-Gradient Correlation Histogram*

Rather than computing two vectors and appending them together as described above, this vector is computed by measuring the correlation between color and gradient direction in the image. It is computed by assigning a bin to every possible color and gradient direction pair and then summing into those bins the gradient magnitudes of pixels that have the corresponding color-gradient direction pair. The vector $\vec{v}$ can be represented as follows:

$$\vec{v} = \frac{[m_1, \cdots, m_N]}{\sum_{i=1}^{N} m_i}$$

$$m_i = \sum_{\substack{p \in G \wedge \\ C(p) \times N_\Theta + \Theta(p) = i}} l_p$$

where the dimension of the vector is decided by $N = N_R \times N_G \times N_B \times N_\Theta$ and $N_\Theta \approx N_R \times N_G \times N_B$.

4.4 Random Vector

This is simply a vector of random numbers of given dimension whose evaluation results serve as a basis of observation. The vector can be represented like so:

$$\vec{v} = [r_1, \cdots, r_d]$$
$$r_i = rand(0, 1)$$

where $rand(0,1)$ simply a function that returns a random real number from the given range.

5. EXPERIMENT RESULTS

The experiment is performed using a collection of photos downloaded from the Flickr website. We selected 22 keywords and used use each of them to retrieve 100+ photos though photo tag search. It is worth nothing that some of the keywords used, such as lily and daisy, returns visually similar photos, which makes it challenging to clustering.

In all experiments described below, the number of units in the SOM is set to 256×256, which is large enough to ensure photos with distinct feature vectors be mapped to unique locations in 2D canvas. The number iterations use in SOM training is fixed at 30, which is sufficient for the process to converge.

5.1 Photo Distribution Measure

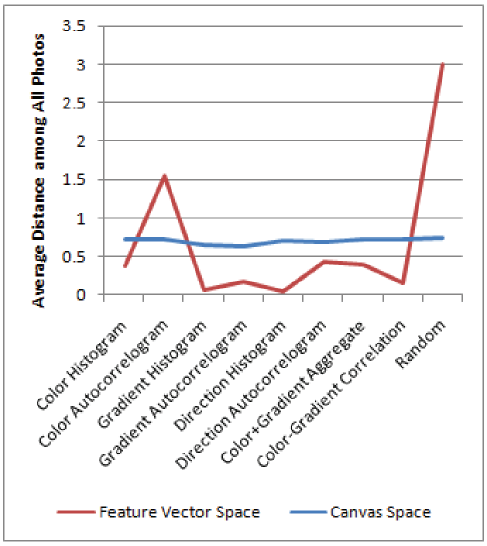


Figure 1: The average span of all photos on 2D canvas and the average feature vector distance in the feature space

We first evaluate whether the SOM training approach can evenly distribute photos across the 2D canvas under different feature vectors settings. Figure 1 plots the average span of all photos on the 2D canvas, the measure F, as well as the average feature vector distance in the vector space. It confirms that, when different types of feature vectors are used, the average distances among feature vectors differ dramatically. However, the average span of photos' locations on the 2D canvas stays at about 0.71, close to the average span of randomly positioned photos. This suggests the SOM training algorithm can effectively map photos across the 2D canvas, regardless of the original feature vector distribution. Browsing photos using the organizing results also shows that photos are evenly distributed (see Figure 2).

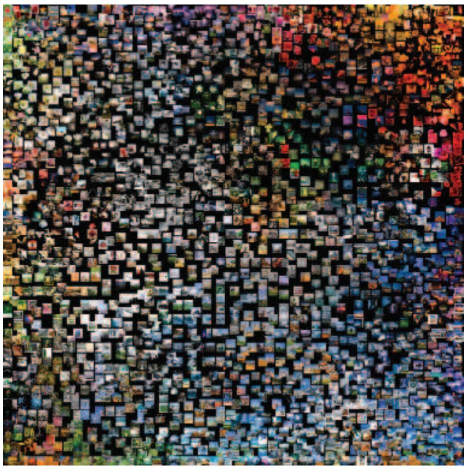


Figure 2: The photo organizing result obtained for all 2200+ photos using color autocorrelogram feature vector

5.2 Performance of Different Feature Vectors

Since not all the photos found by Flickr search are visually describing the keyword used, we manually selected 10~15 photos for each category and use them as the ground truth classification for evaluating the different feature vectors. Figure 3 shows the average effectiveness measured, E, for different feature vectors and under different vector size settings. The following can be observed from the evaluation:

- When the size of the feature vector is limited to four dimensions, the two color-base approaches offer the best results. However, increasing the feature vector size does not necessarily improve the clustering.
- The gradient-based approaches generally perform better with larger feature vector settings.
- When the feature vector size is larger than four, the gradient histogram is outperformed by the gradient direction histogram and the gradient autocorrelogram is outperformed by the gradient direction autocorrelogram. This seems to suggest that the gradient magnitude information does not help much in depicting visual contents of photos.
- As expected, the best performances are obtained using the two hybrid approaches. Among the two approaches, the color-gradient correlation feature vector performs better at higher dimensions and the simple aggregated vector performs better at lower dimensions.

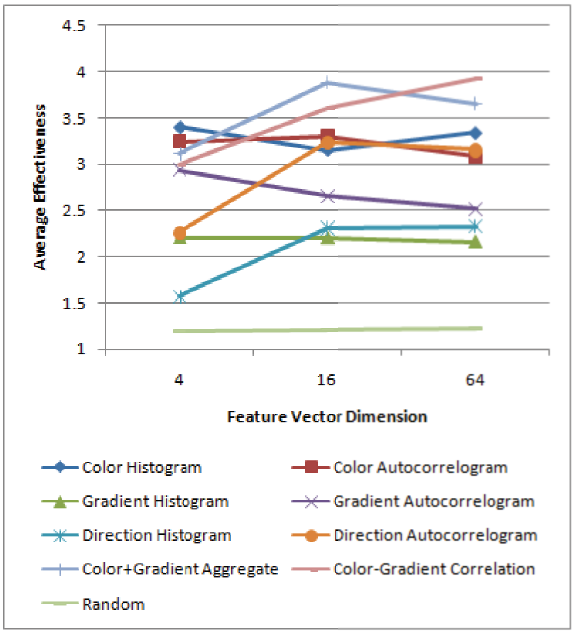


Figure 3: The effectiveness of different feature vectors under different vector dimension settings

5.3 Performance on Different Image Categories

Figure 4 shows the effectiveness obtained for different image categories using four selected feature vectors. The results show that:

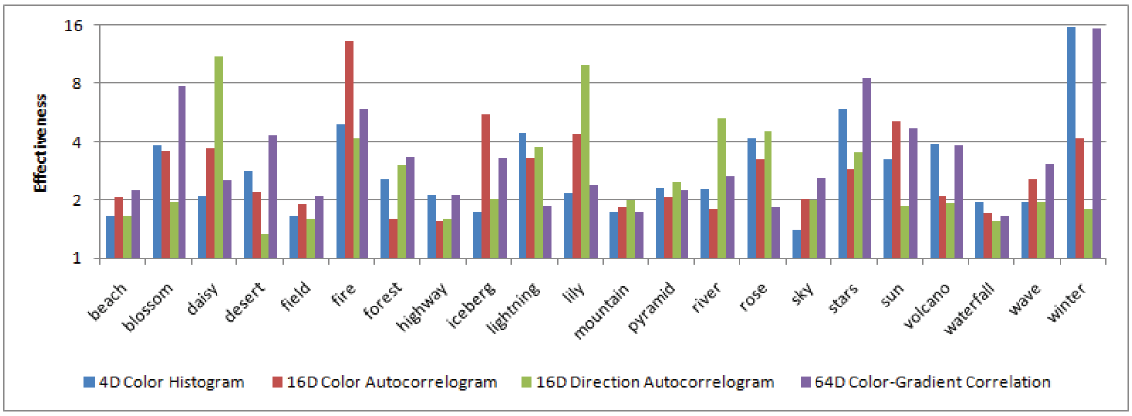
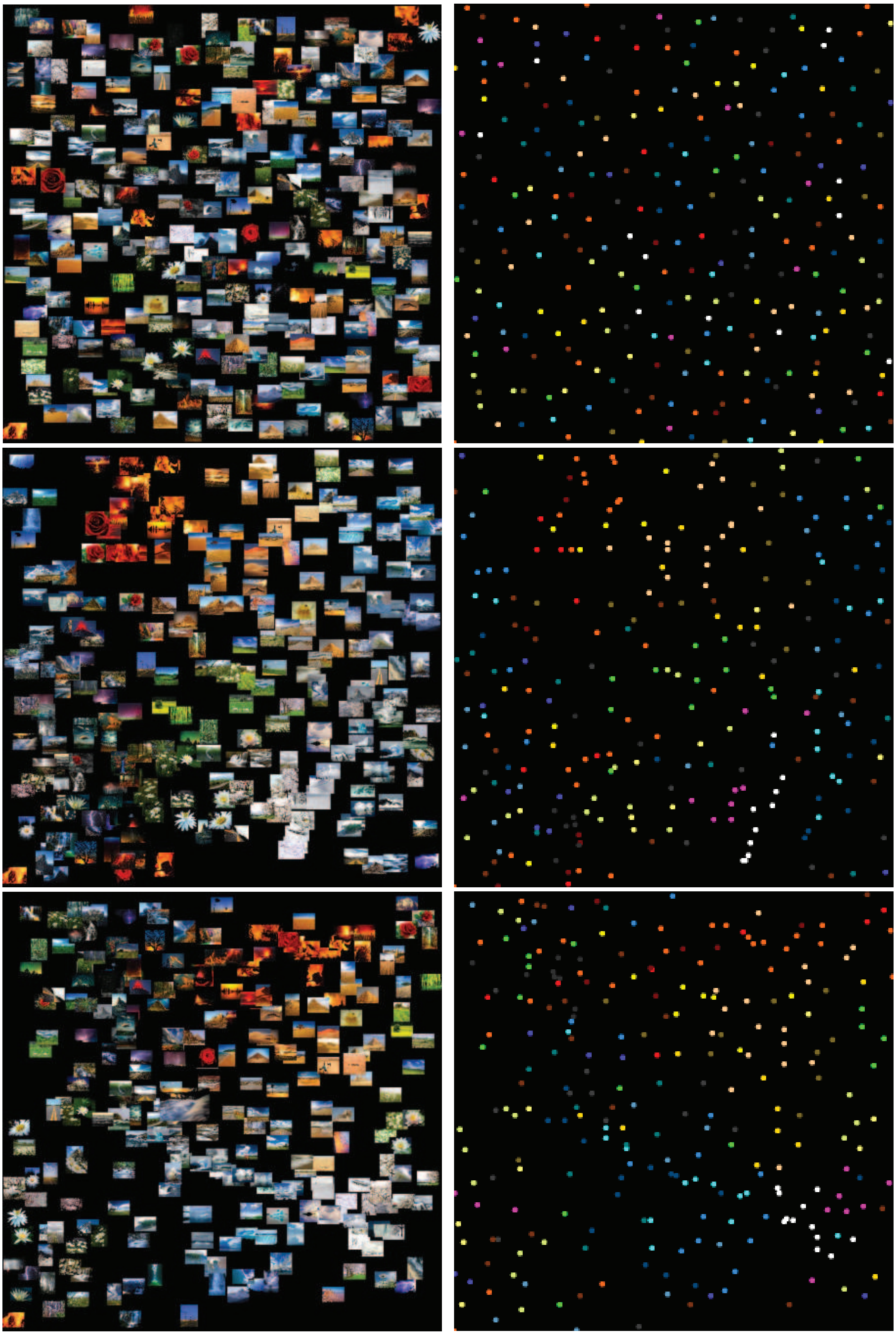


Figure 4: The effectiveness for four different feature vectors on different categories.



(a) 64 dimensional random vector

(b) 64 dimensional color histogram vector

(c) 64 dimensional color-gradient correlation vector

Figure 5: Photo organizing results obtained. Left column shows all the photos and right column shows the distribution of different categories using different colors. Please note that the 2D virtual canvas is wrapped around for better browsing experience and hence similar photos may be placed near boundaries of the opposite sides.

- Using only four numbers, the color histogram can effectively group together photos of fire, lightning, rose, star and winter. This can be attributed to the distinct color distributions of these photos.
- The color autocorrelogram performs the best on the fire, iceberg, and sun categories. This may due to the unique neighbor color pattern existed in photos of these categories.
- The gradient direction autocorrelogram is very effective for grouping daisy and lily photos. These photos contain detailed textures and allow the gradient direction autocorrelogram vector to extract distinct gradient direction pattern. However, it does poorly on desert and winter photos where the image gradient is not as prevalent.
- With combined color and gradient direction information, the hybrid color-gradient correlation approach performs the best for blossom, desert, forest, sky, stars, and wave photos and also performs well in categories such as fire, sun and winter.
- Some categories are hard to distinguish from others since they are visually similar. For example waterfall photos may appear similar to wave or river photos. Beach and field photos contain large portion of sky, which can influence the feature vectors generated.

Figure 5 shows the photo organizing results generate using three different feature vectors. As expected, photos are placed at random locations when the random feature vector is used for training. The result obtained using color-gradient correlation feature vector is noticeable better than the one generated with color histogram, which is consistent with the effectiveness values measured.

6. CONCLUSIONS

This paper studies the 2D similarity-based image organizing and browsing problem, the objective of which is to position photos based on their visual similarities on 2D virtual canvas for a better photo browsing experience. We believe that solving the 2D similarity-based photo organizing problem has important and practical applications. For example, operating systems can adopt this technique as an option to arrange photos in a folder by their visual contents, rather than the visually meaningless file names or sizes.

The presented paper extends our previous work [10] in both photo organizing and browsing directions. In terms of organization, the new application allows the users to select from eight different ways of extracting feature vectors for organizing photos, whereas our previous approach only supports two. In terms of photo browsing, here we propose to pre-order all the photos based on both multi-resolution SOM and a dithering matrix. When only a subset photos can be displayed due to screen resolution, the pre-computed order allows a consistent set of photos to be selected. This solves the photo flickering problem of our previous approach, which uses K-means algorithm to dynamically select photos for display [10].

The main focus of this paper however is on the evaluation of the effectiveness of different feature vectors for 2D photo organization. A performance metric is proposed to measure how well photos belonging to the same category are grouped together on the 2D canvas. The evaluation reveals the pros and cons of different feature vectors, which can be a useful guide for developing new feature vectors.

Much future work can be done along this direction since we only implemented and tested simple feature extraction approaches in this paper to ensure that photos can be organized at interactive speeds. Whether or not more complex feature extraction approaches can help to group visually similar photos into tighter areas is worth investigation.

7. ACKNOWLEDGMENTS

This research is supported by NSERC and Memorial University of Newfoundland.

8. REFERENCES

[1] M. Aleman-Flores and L. Alvarez-Leon, "Texture Classification through Multiscale Orientation Histogram Analysis," in *Scale Space Methods in Computer Vision*, 2003, pp. 479-493.

[2] R. Datta, D. Joshi, J. Li, and J. Z. Wang, "Image Retrieval: Ideas, Influences, and Trends of the New Age," *ACM Computing Surveys*, vol. 40, no. 2, 2008.

[3] D. Heesch, "A survey of browsing models for content based image retrieval," *Multimedia Tools and Applications*, no., 2008.

[4] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih, "Image Indexing Using Color Correlograms," *Proc. CVPR*, pp. 762, 1997.

[5] T. Kohonen, *Self-Organization Maps*. Berlin: Springer-Verlag, 1995.

[6] J. Laaksonen, M. Koskela, and E. Oja, "PicSOM – content-based image retrieval with self-organizing maps," *PRL*, vol. 21, no. 13-14, pp. 1199-1207, 2000.

[7] P. Prentis, "Galsom - colour-based image browsing and retrieval with tree-structured self-organising maps," *Proc. International Workshop on Self-Organizing Maps*, 2007.

[8] 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 TPAMI*, vol. 22, no. 12, pp. 1349-1380, 2000.

[9] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: Exploring photo collections in 3D," *Proc. Siggraph*, pp. 835-846, 2006.

[10] G. Strong and M. Gong, "Browsing a large collection of community photos based on similarity on GPU," *Proc. International Symposium on Visual Computing*, pp. 390-399, 2008.

[11] K.-L. Tan, B. C. Ooi, and C. Y. Yee, "An Evaluation of Color-Spatial Retrieval Techniques for Large Image Databases," *Multimedia Tools and Applications*, vol. 14, no. 1, pp. 55-78, 2001.

[12] R. S. Torres, C. G. Silva, C. B. Medeiros, and H. V. Rocha, "Visual structures for image browsing," *Proc. Conference on Information and Knowledge Management*, pp. 49-55, 2003.

[13] X. S. Zhou and T. S. Huang, "Relevance feedback in image retrieval: A comprehensive review," *Multimedia Systems*, vol. 8, no. 6, pp. 1432-1882, 2003.