Temporal Event Clustering for Digital Photo Collections

MATTHEW COOPER,
JONATHAN FOOTE,
ANDREAS GIRGENSOHN,
and
LYNN WILCOX
FX Palo Alto Laboratory
Palo Alto, CA USA

Organizing digital photograph collections according to events such as holiday gatherings or vacations is a common practice among photographers. To support photographers in this task, we present similarity-based methods to cluster digital photos by time and image content. The approach is general, unsupervised, and makes minimal assumptions regarding the structure or statistics of the photo collection. We present several variants of an automatic unsupervised algorithm to partition a collection of digital photographs based either on temporal similarity alone, or on temporal and content-based similarity. First, inter-photo similarity is quantified at multiple temporal scales to identify likely event clusters. Second, the final clusters are determined according to one of three clustering goodness criteria. The clustering criteria trade off computational complexity and performance. We also describe a supervised clustering method based on learning vector quantization. Finally, we review the results of an experimental evaluation of the proposed algorithms and existing approaches on two test collections.

Categories and Subject Descriptors: H.5 [**Information Systems**]: INFORMATION INTERFACES AND PRESENTATION; H.3.1 [**Information Systems**]: INFORMATION STORAGE AND RETRIEVAL—*Content Analysis and Indexing*; *Indexing Methods*

General Terms: Algorithms, Management

Additional Key Words and Phrases: Digital photo organization, temporal media indexing, digital libraries

1. INTRODUCTION

Digital cameras are coming into widespread use, and as a result, consumers are amassing increasingly large collections of digital photographs. There is a growing demand for automatic tools to help manage, organize, and browse these collections. A recent study focused on requirements for these tools [Frohlich et al. 2002]. The authors emphasized the importance of intuitive photo management software capable

Author's address: M. Cooper, FX Palo Alto Laboratory 3400 Hillview Ave. Palo Alto, CA, 94304. Email: cooper@fxpal.com

of supporting a variety of usage scenarios. While a number of tools now exist for consumer photo management, the vast majority feature light tables of thumbnails in chronological order.

Such tools exploit the two essential cues by which users navigate their photos: temporal order and visual content. Intuitively, users associate time and content with the notion of a specific "event." Thus photo collections are often organized according to events, for browsing, retrieval, and sharing selected photos with others. While events are naturally associated with specific times and places, such as a vacation or a child's birthday party, events remain difficult to define quantitatively or consistently. The photos associated with an event often exhibit little coherence in terms of either low-level image features or visual similarity. Consider possible pictures taken during a trip to the beach. The photos could have widely different subjects such as the beach, the ocean, boats, or people. Photos of the same scene will also exhibit considerable variability if taken at different times of day.

Generally, photographs from the same event are taken in relatively close proximity in time. [Graham et al. 2002] reported that organizing photos by time significantly improves users' performance in retrieval tasks. That study attributes a 33% increase in the speed with which users retrieve specific photos to the ability to navigate the photos after time-based clustering. Users also "clearly preferred" the cluster-based organization to a linear temporal ordering without clustering.

Rodden and colleagues have conducted some of the most extensive studies of users' practices for organizing and searching personal print and digital photograph collections [Rodden 2002; Rodden and Wood 2003]. In a survey in 2000, non-digital photographers were asked what features they desired in software for organizing digital photographs. The highest rated feature was "the ability to organize photographs into folders of some kind... when asked how they would use this facility, participants said that they would arrange their photographs according to events, in chronological order." Likewise, the ability to search a photo collection using date/time information was the second highest rated potential search feature. The highest rated feature was search based on available text notes or annotations, which must be manually supplied in contrast to the automatically recorded timestamps.

In a subsequent study of digital photographers' practices, [Rodden 2002] reported that "the most important use of digital photographs is to record holidays or other significant events, and then show the pictures to friends and family." He added that participants organized their photos similarly to their non-digital photos. That is, digital photos were grouped into folders (directories) according to the events they chronicled. For these studies, photo management software developed by AT&T Labs called Shoebox [Mills et al. 2000] was deployed for users to organize their collections into "databases" (i.e. collections) comprised of "rolls" (sub-collections). Users rated the ability to group photos into separate rolls as the most useful feature of Shoebox. The second most useful feature was the ability to give the roll a title. These rolls and their titles were most often associated with one or more events.

The importance of events is also evident in studies of users' practices when searching through their archived photo collections. Three basic types of searches were reported in [Rodden and Wood 2003], ordered in decreasing frequency:

(1) Search for a set of photos from a particular event

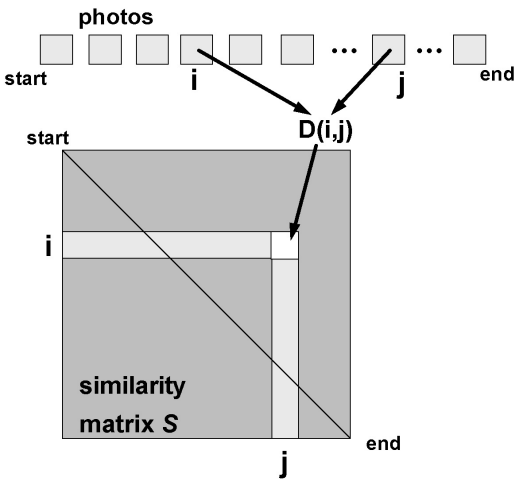


Fig. 1. Embedding photo similarity data in a matrix.

(2) Search for an individual photograph
(3) Search for a set of photographs from different events with a shared attribute such as a certain person.

Notice that each of these tasks is defined relative to an event-based organization of the collection. For the second task, users often first recall the event associated with the specific desired photo to facilitate locating that photo by browsing. This strategy could also apply to the third task, though the studies do not report this specific practice. Finally, although Shoebox included other organizational and search capabilities, including text and spoken annotation, these were not used often for search or organization. Rather, simple time-ordered organization into event-based "rolls" was deemed convenient and sufficiently powerful to satisfy most participants' organizational needs.

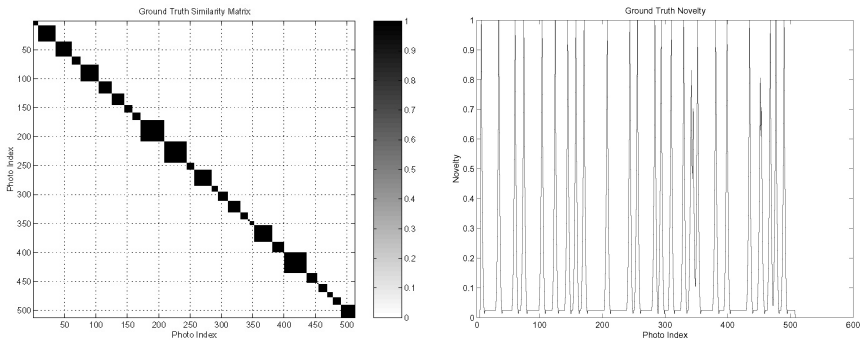
Motivated by these studies, we focus on analyzing the photos' timestamps to automatically organize digital photo collections into event-based clusters. The framework presented herein is general and extensible, allowing metadata and content-based information to be integrated. We formulate event detection as the partitioning of the time interval of the photos' timestamps into contiguous subintervals corresponding to the underlying events. For partition boundaries, we only consider the times at which photos were taken; each photo is a candidate event boundary.

1.1 Similarity analysis

Our approach is based on quantifying the similarity between the photos' timestamps. The first step is to extract and sort the timestamps in a photo collection. Digital photographs typically include metadata, such as the time and date, in a standard image header such as EXIF (EXchangeable Image File [JEIDA 1998]). We quantify temporal similarity by pairwise comparisons of timestamps.

The data comprised of *all* such pairwise comparisons is conveniently visualized in a similarity or affinity matrix such that the (i, j) element of the matrix is the

similarity between the i^{th} and j^{th} photos in time order. This embedding is graphically depicted in Figure 1. As an example, Figure 2(a) shows the similarity matrix generated from the ground truth clustering of 500 photos from our test set. Each photo was stored in an event folder by the photographer. The elements of the matrix are one for photos from the same folder and zero otherwise. The square blocks along the main diagonal of the matrix are the photos grouped in each folder. A checkerboard pattern along the main diagonal indicates the boundary between folders or events. The center of the checkerboard pattern (on the main diagonal of **S**) is the boundary between the photos in the two events. This visualization immediately shows that the photographer-defined (ground truth) events partition the photos contiguously in time. To see this, notice that the matrix does not have rows (or columns) with zero entries between one entries. Each row's elements that equal one (members of the same event) are always connected. We assume that the events are contiguous and each photo belongs to a single event. Thus event detection reduces to locating the event boundaries.



(a) (b)

Fig. 2. Panel (a) shows the ground truth similarity matrix. Panel (b) shows the novelty score computed using a gaussian checkerboard kernel.

To identify the event boundaries, we detect the checkerboards in **S**. The checkerboard patterns indicate two adjacent groups of photos with high intra-group temporal similarity and low inter-group similarity. To locate the event boundaries, we adapt a media segmentation algorithm [Foote 2000], founded on the computation of a photo-indexed novelty score as detailed in Section 3. Peaks in the novelty score correspond to likely event boundaries. As an example, the novelty score computed from the matrix of Figure 2(a) appears in Figure 2(b). Because the duration of an event can be anywhere from hours to weeks, we examine similarity at multiple scales using an indexed family of temporal similarity measures. Thus each photo is associated with a novelty score at each scale. These novelty scores form the basis for subsequent clustering algorithms.

1.2 Overview

The paper is organized as follows. Section 2 reviews related work. Section 3 describes the calculation of the photo-indexed novelty scores used as features for event

detection. Using these features, Section 4 presents a supervised algorithm for event detection. We train a learning vector quantizer (LVQ) to classify each photo's features as either an event "boundary" or "interior" (non-boundary). Although the novelty features include temporal information in a local neighborhood in the photo collection, the LVQ classifies each photo independently. Section 5 describes an unsupervised algorithm using the multi-scale features. Peaks in the novelty scores are detected at each scale. A hierarchical set of event boundaries is constructed by processing the boundary lists from coarse scale to fine. The photo clusterings at each scale are then quantitatively compared to select a "best" scale, and the corresponding boundary list provides the final event clustering. We also present a version of this algorithm which integrates content-based and temporal similarity using a simple heuristic.

A drawback of this algorithm is its quadratic complexity in the number of photos. In Section 6, we expand and update [Cooper et al. 2003] by introducing two reduced complexity variations. Both are based on efficient techniques for selecting a final subset of event boundaries from the set detected across all considered scales. The first approach is based on the Bayes information criterion (BIC) and the second is based on dynamic programming. Both methods achieve competitive performance levels at reduced computational cost. In Section 7, we present experimental results comparing the proposed approaches and several competing methods on two test collections of digital photos classified into meaningful events by the photographers. The paper concludes with a summary discussion.

Our temporal event clustering is unsupervised and automatic and its performance approximates that of hand-tuned techniques (i.e. algorithms with thresholds that are manually set to optimize performance). The similarity-based framework presented below is very general. It can integrate content-based features and relevant metadata, and the multi-scale novelty features and analysis can be applied to text, audio, and video stream segmentation. Also, the formulation based solely on temporal similarity can be used to analyze any timestamped data collection.

2. RELATED WORK

Automatic digital photo organization has received increased attention in recent years. The algorithms in [Graham et al. 2002; Platt et al. 2003] group photos using an adaptive local threshold applied to the inter-photo time interval. Researchers at Kodak segment events by clustering time differences using the two class K-means algorithm and content-based post-processing [Loui and Savakis 2003]. Prior to the K-means clustering, a warping is applied to the inter-photo time differences. The system also includes content-based clustering to determine sub-events. The content-based processing is used for event clustering for photos without timestamps. All time differences in the cluster with the greater mean are labelled as event boundaries. The STELLA system includes a semi-automatic algorithm for content-based event clustering using image sequence (within a roll of film) information rather than timestamps [Jaimes et al. 2000]. In [Mojsilovic et al. 2002], semantically-motivated content-based features were developed for image indexing and retrieval without the use of metadata. Lim *et al.* use machine learning and an event taxonomy to model visual events [Lim et al. 2003]. New photos are compared to existing event models

for event-based organization and retrieval. Gargi presents a bottoms-up clustering approach that characterizes event boundaries as being separated by relatively large time intervals followed by a "burst" or local increase in the frequency of photos taken [Gargi 2003]. The paper also includes an analysis of the distribution of photos' timestamps, arguing that individual photographers collections can be characterized by their fractal dimension. [Naaman et al. 2004] addresses the organization of digital photos including both temporal and location information (e.g. GPS). While most current cameras and photographs do not include GPS, future cameras (and cell-phone cameras) will be able to supply location information. This system first uses a variant of the temporal clustering algorithm in [Graham et al. 2002] to over-segment the collection. In a second pass, a clustering algorithm is applied using the physical rather than temporal distances. Post-processing based on the geographical clusters determines the final photo organization.

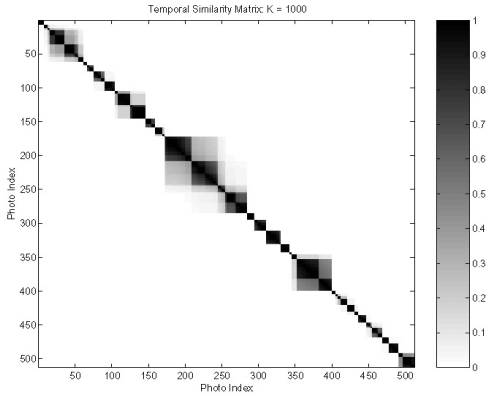
Our work is closer in spirit to scale-space analysis [Witkin 1984] and its application to the segmentation of text and video streams in [Slaney et al. 2001]. In scale-space analysis, difference features are extracted from a data set and examined after smoothing with Gaussian kernels of varying standard deviation. The multiple smoothing filters reveal boundaries at the varying scales. The boundaries are detected and traced back from fine to coarse scale. Final segment boundaries are selected according to the strength and extent of the peaks over the scales. This information can be used to construct a final flat or hierarchical segmentation.

In this paper, we focus primarily on temporal organization of photo collections at multiple scales. Our approach, detailed below, is fully automatic and requires no thresholds or training. Unlike [Graham et al. 2002; Platt et al. 2003], temporal similarity is assessed at multiple scales, and the similarity measure is calculated between *all pairs* of points in local temporal neighborhoods (including photos that are not adjacent in time order). At each scale, we compute a correlation-based score to determine locally novel data points between two adjacent groups of homogenous features that exhibit low inter-group similarity. To select a final set of event boundaries, we determine a "best" scale for the event segmentation over the entire collection of photos. Unlike [Slaney et al. 2001], the scale varies *in the similarity measure*, used to quantify inter-photo temporal similarity. We use the same kernel at every scale to compute the novelty features of Section 3.2. Finally, our algorithm does not require segment boundaries to be "traced back" from smaller scales to larger scales. Rather, we use the methods of Section 6 to assess the quality of the clusterings associated with the different sets of boundaries. Clustering at multiple resolutions also enables flexible user interfaces that allow users to organize their photo collections at different time scales.

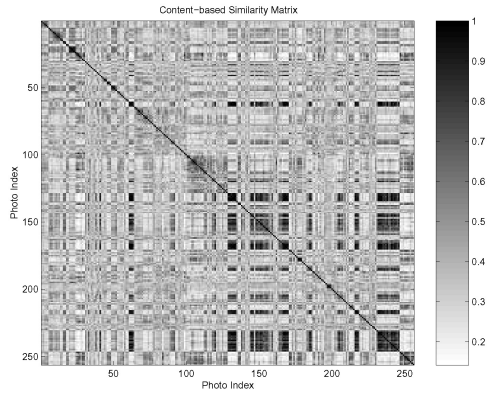
3. FEATURE EXTRACTION

For each photo, the EXIF headers are processed to extract the timestamp (if EXIF information is not available, we rely on the modification time of the digital image file). The N photos in the collection are then ordered in time so the resulting timestamps, $\{t_n : n = 1, \cdots, N\}$, satisfy $t_1 \leq t_2 \leq \cdots \leq t_N$. Throughout, we index the timestamps and the rows and columns of the similarity matrices by photo (in time order), not by absolute time. This differs from [Foote 2000], because the time

difference between indices (photos) is non-uniform. Each photo is represented by its scalar timestamp.



(a)



(b)

Fig. 3. Panel (a) shows a temporal similarity matrix computed for 256 digital photos. Panel (b) shows the content-based similarity matrix calculated from low frequency DCT features and the cosine similarity measure.

3.1 Distance matrix embedding

Figure 3 shows two similarity matrices computed from 256 photos. The photos belong to 11 contiguous event clusters (as grouped by the photographer). The matrices are computed by comparing the features from all possible pairs of photos. The resulting similarity data is embedded in the similarity matrix as in Figure 1. Specifically, the (i, j) element of the matrix quantifies the similarity between the i^{th} and j^{th} photos, ordered chronologically. Figure 3(a) shows the matrix with elements

$$\mathbf{S}(i,j) = \exp\left(-\frac{|t_i - t_j|}{1000}\right) \quad ,$$

where t_i and t_j are the timestamps in minutes of photos i and j, respectively. The darker blocks of high similarity along the main diagonal indicate groups of photos with similar timestamps. The matrix provides a reasonably clear visualization of the temporal structure of the photos.

Figure 3(b) shows the corresponding content-based similarity matrix. The matrix is computed by comparing low frequency discrete cosine transform (DCT) coefficients from each photo using the cosine distance measure:

$$\mathbf{S}_C(i,j) = \frac{< \mathbf{v}_i, \mathbf{v}_j >}{\|\mathbf{v}_i\|\|\mathbf{v}_j\|} \quad , \tag{1}$$

where $\mathbf{v}_i$ denotes the DCT features of the i^{th} photo. Far less structure is evident in $\mathbf{S}_C$, compared to the matrix of panel (a). As illustrated here, content-based image similarity is generally less reliable for photo clustering and event detection than metadata.

We use a multi-scale approach to assess the temporal structure in the photo collection. We construct a family of $N \times N$ similarity matrices according to

$$\mathbf{S}_K(i,j) = \exp\left(-\frac{|t_i - t_j|}{K}\right) \quad . \tag{2}$$

The parameter K controls the sensitivity of the similarity measure. For calculation, the units of K and the timestamps are minutes. By varying K, we assess the similarity between the timestamps at differing granularities. The top row of Figure 4 shows similarity matrices computed using (2) for $K = 10^4, 10^5$ minutes. The matrices for larger values of K exhibit coarser clusterings of the photos' timestamps. For smaller K, finer dissimilarities between groups of timestamps become apparent.

3.2 Computing the novelty scores

In Figure 2(a), the event clusters are visible as dark square blocks on the main diagonal. The boundaries between the event clusters are the centers of checkerboard patterns along the main diagonal. To identify the cluster boundaries between groups of similar photos, we traverse the diagonal and calculate a photo-indexed novelty score, following [Foote 2000]. We seek the centers of the checkerboards; each corresponds to the boundary between two adjacent groups of photos each with high temporal *intra-cluster similarity*. The off-diagonal squares of the checkerboard indicate low temporal *inter-cluster similarity*. The novelty score quantifies local inter-cluster and intra-cluster similarity using a matched filter approach. We correlate a Gaussian-tapered checkerboard kernel, denoted g, along the main diagonal of each $\mathbf{S}_K$ to calculate the photo-indexed novelty score

$$\nu_K(i) = \sum_{l,m=-\ell}^{\ell-1} \mathbf{S}_K(i+l, i+m) g(l,m) \quad . \tag{3}$$

Throughout, $\ell = 6$, so that the kernel is 12×12. The bottom row of Fig. 4 shows the novelty scores computed for $K = 10^4, 10^5$ minutes. While the matrices reveal structure at different resolutions, the peaks in the corresponding novelty scores comprise a set of cluster boundaries between contiguous groups of similar photos. The boundaries are identified by simple analysis of each novelty score's first difference.

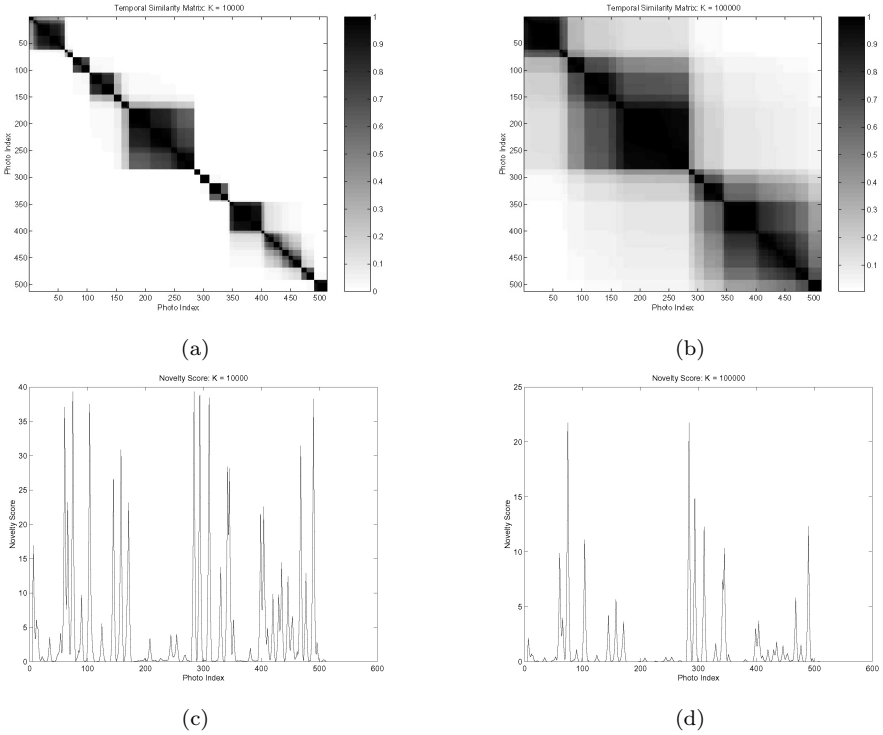


Fig. 4. The left column shows the similarity matrices $\mathbf{S}_K$ for $K = 10000$ (a) and $K = 100000$ (b) minutes. Panels (c) and (d) show the corresponding novelty scores computed using a Gaussian checkerboard kernel.

4. SUPERVISED EVENT CLUSTERING

In this section, we describe a supervised algorithm for event clustering based on a LVQ [Kohonen 1989]. Here, we assume that the multi-scale novelty features can be used to distinguish photos at event boundaries from the remainder of the collection. While we focus here on the LVQ, we note that this general approach can be used with numerous supervised classification techniques. Let K take M values: $K \in \mathcal{K} \equiv \{K_1, \cdots, K_M\}$. Define the $M \times N$ matrix $\mathcal{N}(j,i) = \nu_{K_j}(i)$. We associate photo i with its novelty scores at each scale,

$$\mathcal{N} \equiv \begin{bmatrix} \mathcal{N}_i & \cdots & \mathcal{N}_N \end{bmatrix} \quad \text{where} \quad \mathcal{N}_i = \begin{bmatrix} \nu_{K_1}(i) \\ \vdots \\ \nu_{K_M}(i) \end{bmatrix} \quad . \tag{4}$$

We expect that event boundaries will correspond to local maxima in the novelty scores at a range of scales. At the same time, typical photos in the interior of an event will have low novelty scores (at all scales). The LVQ is designed to provide effective event detection by exploiting these class differences.

Learning vector quantization uses positive and negative examples to select the codebook vectors. In the training phase, a codebook is calculated using an iterative

procedure. At each step, the nearest codebook vector to each training sample is determined, and it is shifted towards the training sample if they are members of the same class, and away from the training sample otherwise. Specifically, if c denotes the index of the nearest codebook vector $\mathcal{M}_c$ to the training sample $\mathcal{N}_x$, then $\mathcal{M}_c$ is updated at iteration $t+1$ as,

$$\mathcal{M}_c(t+1) = \begin{cases} \mathcal{M}_c(t) + \alpha(t)(\mathcal{N}_x - \mathcal{M}_c(t)) \\ \qquad \text{if } \mathcal{N}_x \text{ and } \mathcal{M}_c \text{ are in the same class,} \\ \mathcal{M}_c(t) - \alpha(t)(\mathcal{N}_x - \mathcal{M}_c(t)) \\ \qquad \text{if } \mathcal{N}_x \text{ and } \mathcal{M}_c \text{ aren't in the same class.} \end{cases} \tag{5}$$

α is a scalar between zero and one that decreases with t.

Here, the LVQ codebook discriminates between the two classes "event boundary" and "event interior." We calculate the LVQ from a labelled subset of the columns of $\mathcal{N}$. In classification, the LVQ takes in the novelty data $\mathcal{N}_i$ for photos not belonging to the training set and returns the estimated class membership. We construct the LVQ and perform testing using LVQ PAK [Kohonen et al. 1992]. The codebook vectors for each class are used for nearest-neighbor classification [Duda and Hart 1973] of the novelty features for each photo in the test set.

While the approach is non-parametric and discriminative, a key disadvantage is that the decisions for each photo are independent. For example, there are no priors or constraints imposed that prevent two consecutive photos from both being classified as event boundaries, although this is unlikely in practice. An advantage of supervised techniques is that a separate LVQ can be trained for each photographer to capture any user-specific habits in camera use and preferences in event definition. For instance, different photographers may consider a two-week vacation to be either a single event or multiple events. The LVQ will be likely to accommodate such a preference if events with appropriate lengths are well represented in the training data for each photographer.

ALGORITHM 1. **[LVQ-based Photo Clustering]**

(*1*) *Calculate novelty features from labelled sorted training data for each scale* $K \in \mathcal{K}$*:*
  i. Compute the similarity matrix $\mathbf{S}_K$ *using* (2).
  ii. Compute the novelty score ν_K *of* (3).

(*2*) *Train LVQ using the iterative procedure of* (5)*. Note these two steps ((1) and (2)) can be completed off-line.*

(*3*) *Calculate novelty features for the testing data for each* $K \in \mathcal{K}$
  i. Compute the similarity matrix $\mathbf{S}_K$ *using Eq.* (2).
  ii. Compute the novelty score ν_K *of Eq.* (3).

(*4*) *Classify each test sample's novelty features* $\mathcal{N}_i$ *using the LVQ codebook and the nearest-neighbor rule.*

5. UNSUPERVISED EVENT CLUSTERING

Next, we present three unsupervised approaches to event detection. The first is based on scale-space analysis of the raw timestamp data. The second algorithm

processes the multi-scale novelty features of Section 3.2. The final algorithm processes novelty features extracted from similarity matrices that combine temporal and content-based features.

5.1 Scale-space analysis

Scale-space analysis [Witkin 1984] is a technique for assessing structure at multiple scales in a data set. Later, we compare the results of analysis based on the multi-scale novelty features above, with more traditional scale-space features. For the comparison, we operate on the raw timestamps, $T_0 = [t_1, \cdots, t_N]^T$ so that $T_0(i) = t_i$. Gaussian filters of varying standard deviation are applied to the sequence of timestamps to generate an indexed set of signals

$$T_\sigma(i) = T_0(i) * \gamma(i, \sigma) \quad , \tag{6}$$

$$= \frac{1}{\sqrt{2\pi\sigma^2}} \sum_{j=-G}^{G} T_0(i-j) \cdot e^{-\frac{j^2}{2\sigma^2}} \quad , \tag{7}$$

where $2G + 1$ is the extent of the filter γ. Arranging these filtered signals in rows creates a two-dimensional representation: $T(\sigma, i) = T_\sigma(i)$. Peaks in the derivative with respect to i (the photo index) are calculated for each value of σ, indicating segment boundaries. Peaks are "traced" from the larger values to the smaller values of σ, and can be analyzed to assess the importance of the corresponding segment boundary. For event clustering, we use the following algorithm in the experiments discussed later. The form of (7) resembles the kernel correlation of (3). The difference is that the scale parameter is embedded in the similarity data in (3). Additionally, the similarity matrix includes comparisons between features from *both non-adjacent and adjacent* pairs of photos. For the comparison of Section 7, we do not include a final step for peak selection. We use $G = 10$ and manually threshold each T_σ to maximize performance. Various criteria have been proposed to select a final set of peaks in scale space [Leung et al. 2000].

ALGORITHM 2. **[Scale-space Photo Clustering]**

(*1*) *Extract timestamp data from photo collection:* $\{t_1, \cdots, t_N\}$

(*2*) *For each* σ *in descending order*

- *i. Compute* T_σ *as in* (7).
- *ii. Detect peaks in* T_σ*, tracing peaks from larger to smaller scales (decreasing* σ*).*

5.2 Temporal similarity analysis

For temporal analysis, we begin with the family of novelty measures of (3) computed for each $K \in \mathcal{K}$. We first locate peaks at each scale by analysis of the first difference of each ν_K, proceeding from coarse scale to fine (decreasing K). We threshold detected peaks as a function of the maximum novelty for a data-independent approach. The maximum possible novelty score is determined by the similarity measure (which has maximum one here) and the kernel correlated along the main diagonal of the similarity matrix. To build a hierarchical set of event

boundaries, we include boundaries detected at coarse scales in the boundary lists for all finer scales.

5.3 Combining time and content-based similarity

We also implemented a variant of this method which jointly processes content-based features and the photos' timestamps. In particular, we construct a content-based matrix $\mathbf{S}_C$ using low frequency DCT features and the cosine distance measure of (1). One possibility is to use a (piecewise) linear function of the inter-photo time difference to combine $\mathbf{S}_C$ with each of the $\mathbf{S}_K$ of (2):

$$\mathbf{S}_K^{(J)}(i,j) = \begin{cases} \mathbf{S}_K(i,j) & \text{if } |t_i - t_j| > 48 \text{ hours} \\ \alpha \mathbf{S}_K(i,j) + (1-\alpha)\mathbf{S}_C(i,j) & \text{otherwise.} \end{cases} \quad (8)$$

$$\text{where } \alpha = \frac{|t_i - t_j|}{48 \text{ hours}}$$

Again, K indexes the family of similarity measures per (2). In this case, $\mathbf{S}_K^{(J)}$ relies less on content-based similarity as the inter-photo time difference grows. Alternately, we combine the temporal and content-based similarity measures to build the family of matrices, $\mathbf{S}_K^{(J)}$ according to

$$\mathbf{S}_K^{(J)}(i,j) = \begin{cases} \mathbf{S}_K(i,j) & \text{if } |t_i - t_j| > 48 \text{ hours} \\ \max(\mathbf{S}_C(i,j), \mathbf{S}_K(i,j)) & \text{otherwise.} \end{cases} \quad (9)$$

This heuristic emphasizes temporal similarity, which is generally more reliable for organization. However, image similarity can dominate for photos with sufficient temporal proximity and high content-based similarity. In our experience, the method of (9) has consistently outperformed that of (8), and we use (9) in the comparative evaluation below [1]. For the experiments, we substitute $\mathbf{S}_K^{(J)}$ into step 2(a) of Algorithm 3. In future work, we hope to examine other techniques for combining content-based and temporal information for photo organization. In addition, there are numerous other content-based features worth investigating in this framework.

6. CLUSTERING GOODNESS CRITERIA

The peak detection at each scale $K \in \mathcal{K}$ results in a hierarchical set of candidate boundaries. From these, a subset must be selected to define the final event clusters. In this Section we consider three different automatic approaches for determining this subset. As we shall see, the various techniques tradeoff computational complexity and performance. All three techniques can be applied to the set of boundaries determined from either temporal similarity analysis, or combined temporal and content-based similarity analysis. In our experimental evaluation we include both of these variations for each method.

[1] Using (8), the results for Collection I of Table III are precision = 0.8, recall = 0.62, F-score = 0.7. For Collection II of Table IV, precision = 0.74, recall = 0.78, F-score = 0.76. These results can be compared to those for the system JS-C corresponding to (9) in Tables III and IV.

6.1 Similarity-based confidence score

The similarity matrix provides a natural means to assess the quality of the clustering implied by the boundaries at a given scale. For this, we calculate a confidence measure from the average intra-cluster similarity and the inter-cluster dissimilarity of the data. Denote the detected boundaries at each level, $\mathcal{B}_K = \{b_1, \cdots, b_{n_K}\}$, indexed by photo: $\mathcal{B}_K \subset \{1, \cdots, N\}$. For convenience, assume that $b_1 = 1$ and $b_{n_K} = N$ for all K. We then compute the confidence score

$$C_S(\mathcal{B}_K) = \sum_{l=1}^{|\mathcal{B}_K|-1} \sum_{i,j=b_l}^{b_{l+1}} \frac{\mathbf{S}_K(i,j)}{(b_{l+1}-b_l)^2} \; - \; \sum_{l=1}^{|\mathcal{B}_K|-2} \sum_{i=b_l}^{b_{l+1}} \sum_{j=b_{l+1}}^{b_{l+2}} \frac{\mathbf{S}_K(i,j)}{(b_{l+1}-b_l)(b_{l+2}-b_{l+1})} \quad . \quad (10)$$

The first term above quantifies the average intra-cluster similarity between the photos within each cluster. The second term quantifies the average inter-cluster similarity between photos in adjacent clusters. By negating this term, the confidence measure thus combines each cluster's average similarity and the dissimilarity between adjacent clusters. Fig. 5 illustrates the idea graphically. The within-class similarity terms are the means of the terms of darker regions along the main diagonal. The between-class terms are the means of the off-diagonal gray regions.

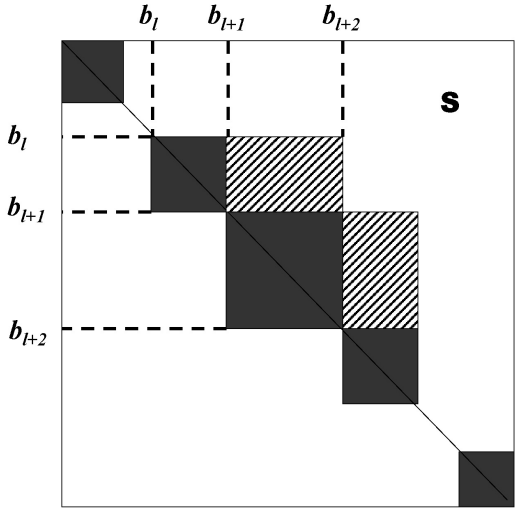


Fig. 5. Computing a confidence score for clustering. The dark regions represent within-cluster similarity, while the gray regions represent between-cluster similarity.

6.2 Boundary selection via dynamic programming

Inspection of (10) reveals that computing the confidence score has quadratic complexity in the number of photos ($O(N^2)$). Thus, we consider two reduced complexity alternatives. Dynamic programming (DP) is an efficient optimization procedure which exploits the optimal solution's structure. Specifically, by associating a cost with each event cluster, we can determine a final partitioning to minimize the total cost. For this, we employ a standard DP procedure for grouping an ordered set of objects due to Fisher [Hartigan 1975]. We begin with the set of peaks detected from the novelty features at *all scales*. Define

$$\mathcal{B} \equiv \bigcup_{K \in \mathcal{K}} \mathcal{B}_K \quad .$$

Generally, $\beta = |\mathcal{B}| \ll N$. To apply the Fisher algorithm, we define the cost of the cluster between photos b_i and b_j to be the empirical variance of the corresponding timestamps:

$$C_F(b_i, b_j) \;=\; \frac{1}{b_j - b_i - 1} \sum_{n=b_i}^{b_j - 1} (t_n - \hat{\mu}_{ij})^2 \quad , \tag{11}$$

$$\text{where} \;\; \hat{\mu}_{ij} \;=\; \frac{1}{b_j - b_i - 1} \sum_{n=b_i}^{b_j - 1} t_n \quad .$$

The algorithm successively builds optimal partitions with m boundaries based on the optimal partition with $m-1$ boundaries. First, optimal partitions are computed with two clusters:

$$E_F(j, 2) = \min_{2 \le i \le j} C_F(1, b_i) + C_F(b_i, b_j) \;, \quad i \le j \le \beta \quad . \tag{12}$$

$E_F(j, m)$ is the optimal partition of the photos with indices $1, \cdots, b_j - 1$ with cardinality m. This procedure is repeated to compute

$$E_F(j, L) = \min_{L \le i \le j} E_F(i, L-1) + C_F(i, j) \;, \quad L \le j \le \beta \quad , \quad 3 \le L \le \beta \quad . \tag{13}$$

The result is a set of optimal partitions with cardinality[2] $3, \cdots, \beta$. A traceback step identifies the boundaries in each of the optimal partitions. As the number of clusters increases, the total cost of the partition decreases monotonically. Various criteria have been proposed for selecting the optimal number of clusters, L^*, based on the total partition cost. We choose

$$L^* \;=\; \underset{2 \le m \le \beta - 1}{\text{ArgMax}}\; g(m) \quad , \tag{14}$$

$$\text{where} \;\; g(m) \;=\; \frac{E_F(\beta, m)}{E_F(\beta, m+1)} \quad . \tag{15}$$

The complexity for computing the costs C_F is quadratic in β, the number of detected peaks in the novelty scores ($O(\beta^2)$). The costs of computing the sample variances of (11) is $O(N)$, but in our experiments, $N < \beta^2$. This is a reduction in computational cost relative to the similarity-based score of (10).

[2]Again we assume that $b_1 = 1$, and $b_\beta = N$.

6.3 BIC-based boundary selection

We present a third approach for determining the final event segmentation from $\mathcal{B}$. This method is based on the Bayes information criterion (BIC) [Schwarz 1978] which is a technique for model order selection. The model order in this context is the number of event clusters. We make a simplifying assumption that timestamps within an event are distributed normally around the event mean. While this assumption is difficult to justify empirically, studies such as [Gargi 2003] have shown that the timestamp distribution may not be easily modelled by a convenient parametric family. We proceed remarking that improved density estimates will enhance the effectiveness of this technique. The basic process is to test each boundary $b \in \mathcal{B}$ to determine if the increase in model likelihood justifies the additional parameters used to describe the additional segment. This results in a simple test for each b_l:

$$L(b_{l-1}, b_l) + L(b_l, b_{l+1}) \gtrless L(b_{l-1}, b_{l+1}) + \frac{\lambda}{2} \log(b_{l+1} - b_{l-1}) \quad . \tag{16}$$

The left hand side is the log-likelihood of the two segment model. The right hand side is the log-likelihood of the single segment model and the penalty term for the additional parameters in the two segment model. λ is the number of parameters required to represent a segment. If the likelihood gain associated with separate models for the two segments exceeds the penalty for the additional parameters, b_l is included in the final event partitioning. In our case λ is two since we describe each segment using the sample mean, $\hat{\mu}_l$ and variance, $\hat{\sigma}_l^2$ of its photos' timestamps:

$$L(b_l, b_{l+1}) = -\frac{b_{l+1} - b_l}{2} \log 2\pi\hat{\sigma}_l - \sum_{n=b_l}^{b_{l+1}} \frac{(t_n - \mu_l)^2}{2\hat{\sigma}_l^2} \tag{17}$$

$$= -\frac{b_{l+1} - b_l}{2} \left(1 + \log(2\pi\hat{\sigma}_l)\right) \quad . \tag{18}$$

To apply the BIC to boundary selection, we employ the same hierarchical coarse-to-fine approach of Section 5.2. At each scale, we test only the newly detected boundaries (undetected at coarser scales) using (16), and add the boundaries for which the left side exceeds the right side. The number of tests is $O(\beta)$ with total computation of the sample means and variances for each segment being $O(N \cdot M)$. This is a significant decrease in computational cost over the score of (10). We summarize the similarity-based event clustering algorithm in Algorithm 3.

ALGORITHM 3. **[Similarity-based Photo Clustering]**

(*1*) *Extract and sort photo timestamps,* $\{t_1, \cdots, t_n\}$.

(*2*) *For each* K *in decreasing order*
   i. Compute the similarity matrix $\mathbf{S}_K$ *using Eq.* (2).
   ii. Compute the novelty score ν_K *of Eq.* (3).
   iii. Detect peaks in ν_K.
   iv. Form event boundary list using event boundaries from previous iterations and newly detected peaks.

(*3*) *Determine a final boundary subset of collected boundaries over all scales considered according to one of the methods in Section 6:*

a. The confidence score of Eq. (10)
b. The DP boundary selection approach
c. The BIC boundary selection approach

6.4 Computational complexity

We review the computational complexity of Algorithm 3. Sorting the N timestamps is $O(N \log(N))$. Computing the entire similarity matrix is $O(N^2)$. Figures 2 and 4 show that the $\mathbf{S}(i,j) = 0$ far from the main diagonal, that is when $|i-j|$ is large. To reduce storage and computation requirements, we need only compute the portion of the similarity matrix around the main diagonal with width ℓ as in (3), reducing complexity to $O(N)$. We use the set of novelty scores computed using matrices with varying values for K as features for photo event clustering. The total cost of determining the candidate boundaries $\mathcal{B}$ is $O(N \cdot M)$, where $M = |\mathcal{K}|$.

Table I. The table documents run times for different size photo collections. The times are in seconds. "No Conf." indicates times for Steps 1 and 2 in Algorithm 3. The remaining column indicate performance of Algorithm 3 using BIC peak selection (BIC), dynamic programming peak selection (DP), and similarity-based peak selection per (10) (Conf.).

Run times (6030 photos total)				
N	No Conf.	BIC	DP	Conf.
500	0.017423	0.016808	0.016808	0.183231
1000	0.034846	0.036077	0.039077	0.563115
2000	0.076923	0.081731	0.128615	1.908038
4000	0.161077	0.179077	0.593769	8.718769
6030	0.271654	0.260231	1.459115	19.594962

The complexity of selecting the final subset of event boundaries is greater. The evaluation of (10) potentially necessitates the computation of the entire similarity matrix, since the extent of events can't be assumed in advance. In the worst case, this includes all N^2 terms of $\mathbf{S}_K$. Because the temporal similarity measure decays exponentially as the time difference increases, we can reduce the complexity using a mask which zeros out elements of the matrix corresponding to photo pairs taken far apart in time. Other heuristics can also be used to construct masks based on the number of photos taken between a pair of photos.

We include representative run times for the temporal-version of Algorithm 3 on a collection of 6030 photos in Table I. The column labelled "No Conf." is the time for steps 1 and 2. The column labelled "Conf." is the time for the entire algorithm with step 3(a). The columns labelled "DP" and "BIC" represent final peaks selection using the DP (step 3(b)) and BIC (step 3(c)) methods of Section 6. Variations of the algorithms were implemented in Java, and the times here were produced using a PC with a 2.66 GHz Pentium 4 processor. As predicted, after doubling the number of photos processed (N), the time for the segmentation step (No Conf.) increases linearly, while including the confidence measure (Conf.) incurs a polynomial cost.

Table II. The algorithms used in our experiments. The second column indicates whether the algorithm is supervised or unsupervised. The third column indicates whether it is manually tuned for our testing. The abbreviations refer to the bar plot of Figure 6.

Algorithm	Supervised?	Automatic?	Reference	Abbreviation
Adaptive Threshold 1	NO	NO	[Platt et al. 2003]	AT1
Adaptive Threshold 2	NO	NO	[Graham et al. 2002]	AT2
Fixed Threshold	NO	NO	-	FT
Scale-space	NO	NO	Section 5.1	SS
LVQ	YES	YES	Section 4	LVQ
Temporal Sim. / Sim.	NO	YES	Section 6.1	TS-C
Temporal Sim. / DP	NO	YES	Section 6.2	TS-DP
Temporal Sim. / BIC	NO	YES	Section 6.3	TS-BIC
Joint Sim. / Sim.	NO	YES	Section 6.1	JS-C
Joint Sim. / DP	NO	YES	Section 6.2	JS-DP
Joint Sim. / BIC	NO	YES	Section 6.3	JS-BIC

As noted earlier, the variants based on DP and the BIC both offer reduced complexity. DP-based peak selection is $O(\beta^2)$ where $\beta < N$. β is governed by the smallest scale in $\mathcal{K}$. Though it's difficult to precisely relate N and β, Table I suggests that DP is roughly ten times faster than the similarity-based peak selection for all values N tested. In more detailed analysis, computing the costs of (11) accounts for two thirds of the time in the DP-based peak selection. The BIC-based peak selection offers more substantial computational savings. As mentioned earlier, we perform $O(\beta)$ tests using (16), and the computation of the sample means and variances is $O(N \cdot M)$, which dominates the total cost. As we shall see, both these efficient alternatives perform competitively with the original similarity-based confidence score.

In practice, the overall runtimes of these methods are fast, even for a large number of photos. Content-based processing, such as thumbnail extraction, is far more computationally expensive than event detection, and for temporal similarity, we process only a single scalar feature per image. In the application of [Girgensohn et al. 2003], we provide a fully automatic solution by using the confidence measure of (10) to select a single scale for the detected event boundaries.

7. EXPERIMENTAL RESULTS

In the previous Sections, we reviewed and presented algorithms for event detection. Here, we compare the event clustering performance of eleven systems on two separate photo collections. Collection I consists of 1036 photos taken over 15 months, and Collection II consists of 413 photos taken over 13 months. All photos had accurate timestamps, and the photos were assigned to meaningful events by the respective photographers. Photos in each event were sequential, and event classifications were used as ground truth for our clustering experiments. Table II enumerates the algorithms used in the evaluation. The first four algorithms in the Table are "hand-tuned" to maximize performance, as quantified by the F-score defined below (Equation 21). The remaining algorithms are fully automatic.

"Adaptive Threshold 1" is based on [Platt et al. 2003] and "Adaptive Threshold

Table III. The table summarizes our experimental results for Collection I.

Collection I			
Algorithm	Precision	Recall	F-score
Adaptive Threshold 1	0.39	1.0	0.56
Adaptive Threshold 2	0.38	1.0	0.55
Threshold	0.72	0.95	0.82
Scale-space	0.86	0.79	0.83
LVQ	0.71	0.80	0.76
Temporal Sim. / DP	0.836	0.807	0.821
Temporal Sim. / BIC	0.754	0.807	0.779
Temporal Sim. / Sim.	0.884	0.807	0.843
Joint Sim. / DP	0.88	0.772	0.822
Joint Sim. / BIC	0.882	0.7895	0.833
Joint Sim. / Sim.	0.9	0.79	0.84

2" is based on [Graham et al. 2002]. The two algorithms are closely related and both compare the time difference between successive photographs to a variable threshold based on the logarithm of the average inter-photo time difference over a local window. Event boundaries occur where the time difference between photos exceeds the threshold. To determine if this worked better than simple thresholding, we skipped their thresholding step and examined the first level of the hierarchy created. The "Threshold" approach is a simple fixed threshold applied to the inter-photo time difference. This threshold is manually adjusted to vary the resulting precision and recall, and optimize the F-score. To test the scale-space approach, we detected boundaries using a simple threshold-based peak detector applied to the filtered signal T_σ for each scale. We employ cross-validation to include the LVQ-based event detector in the comparison. We divide the photos into three (approximately equal) sets for testing. For each test set, we train an LVQ using the remaining data and its ground truth labelling. The results of the three separate tests are combined for comparison with the unsupervised approaches.

The remaining algorithms of Table II are all variants of Algorithm 3. The temporal algorithms all use temporal similarity matrices per (2). The joint similarity algorithms use combined similarity matrices computed per (9). For each of these cases, we consider each choice in Step 3 of Algorithm 3. "Sim.", "DP", and "BIC" refer to peak selection by the similarity-based score of Section 6.1, dynamic programming as in Section 6.2, and the BIC selection as in Section 6.3, respectively.

The precision, recall, and F-score for the detected event boundaries appear in Tables III and IV for each algorithm. These measures are common figures of merit in information retrieval that are also used to assess segmentation performance [Boreczky and Rowe 1996]. Precision indicates the proportion of falsely labelled boundaries:

$$\text{precision} = \frac{\text{correctly detected boundaries}}{\text{total number of detected boundaries}} \quad . \tag{19}$$

Recall measures the proportion of true boundaries detected:

$$\text{recall} = \frac{\text{correctly detected boundaries}}{\text{total number of ground truth boundaries}} \quad . \tag{20}$$

Table IV. The tables summarizes our experimental results for Collection II.

Collection II			
Algorithm	Precision	Recall	F-score
Adaptive Threshold 1	0.42	1.0	0.6
Adaptive Threshold 2	0.29	1.0	0.45
Threshold	1.0	0.85	0.92
Scale-space	1.0	0.83	0.91
LVQ	0.63	0.94	0.76
Temporal Sim. / DP	0.842	0.89	0.865
Temporal Sim. / BIC	0.696	0.89	0.781
Temporal Sim. / Sim.	0.89	0.89	0.89
Joint Sim. / DP	0.89	0.89	0.89
Joint Sim. / BIC	0.842	0.89	0.864
Joint Sim. / Sim.	0.842	0.89	0.864

The F-score is a composite of precision and recall:

$$\text{F-score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad . \tag{21}$$

Notice that the various thresholds are manually adjusted to maximize the F-score for Adaptive Threshold 1, Adaptive Threshold 2, the scale-space, and the simple threshold algorithms. There is no tuning of the LVQ-based method to improve its results. The temporal similarity and joint similarity algorithms are fully automatic.

The adaptive-thresholding algorithms exhibit high recall and low precision on both test sets, even with manual tuning. The LVQ event detector performs better, at least in terms of the F-score. However, it also sacrifices precision for higher recall, and performs slightly worse than the manually tuned threshold and scale-space event detectors. The scale-space and the two similarity-based approaches demonstrate more consistent performance and trade off precision and recall more evenly. As well, the automatic similarity-based algorithms approach the performance of the manually tuned algorithms. The performance on both collections is combined in a weighted average according to the sizes of the two test collections in the bar plot of Figure 6. The temporal-based similarity clustering using peak selection according to Section 6.1 (TS-C) achieves the maximal cumulative F-Score of all the approaches, 0.8568. The accelerated versions of the temporal algorithms also perform at a high level. DP appears to be superior to BIC-based peak selection, particularly with respect to precision.

8. CONCLUSION

In practice, we employ the automatic temporal similarity-based method (Algorithm 3 with the confidence measure of step 3(a)). It has been well received by the pilot users of our application for organizing digital photos [Girgensohn et al. 2003]. For the most part, users did not need to change the automatically detected event boundaries and found it straightforward to assign meaningful titles to the detected event clusters. Figure 7 shows a collection of photos organized by events in the application. The photos appear in time order in the light table. Each event is denoted by a colored label with a name in both the light table pane (right) and

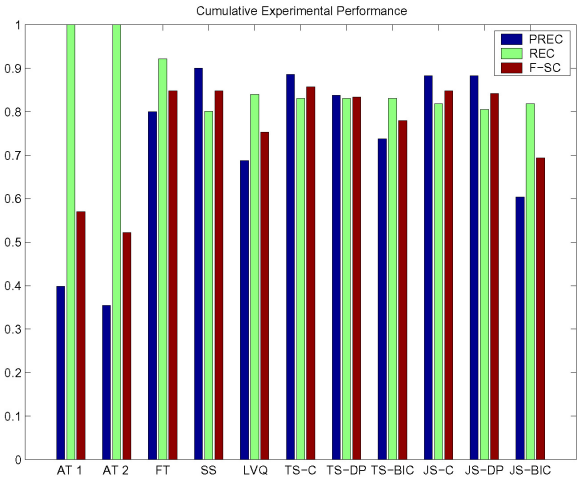


Fig. 6. Experimental comparison of several algorithms for photo event clustering.

the tree pane (left). The events are automatically named using the photos' dates, unless renamed by the user. The photos in the event follow the event label in the rows in the light table. Users may adjust the automatically detected event boundaries by simply dragging and dropping thumbnails onto the desired event label. Additionally, the application allows users to manually set the parameter K in (2) and override the automatic scale selection according to the aforementioned confidence measure. In this way, users can directly select the temporal resolution of the event clustering of their photo collection.

The similarity-based approach has significant advantages over existing techniques. It is very general and allows for the future integration of content-based features or other relevant metadata. Here, we included an initial attempt at combining metadata and content-based features in (9). Other heuristics, weighting schemes, or combinations of multiple similarity measures can also be used to integrate the heterogenous features and metadata describing the photos for automatic organization. While existing approaches typically only consider the similarity between adjacent photos (such as comparing their time difference to a threshold), the novelty measure of (3) is based on similarity comparisons between *all possible* photo pairs in a local neighborhood. Additionally, our approach does not rely on preset thresholds or restrictive assumptions and should generalize better to different image collections. As photo collections with location information become available, we hope to extend our system to combine temporal similarity, content-based similarity, and location-based similarity. [Naaman et al. 2004] uses location information to post-process a temporal clustering. Certainly, location information could be readily used in place of timestamps in the DP or BIC-based peak selection of Section 6. Performance using the BIC in this manner will greatly improve with the accuracy with which the distribution of photographs' locations can be modelled and parameterized.

We have presented several approaches to automatic event clustering for digital photo collections using a general framework based on quantitative inter-photo sim-

Fig. 7. This screen shot shows a user adjusting the results of the automatic event detection in our photo organization application. The user need only drag the thumbnail onto the label for the desired event.

ilarity analysis. Multi-scale similarity features are calculated and used to construct a hierarchical set of event boundaries. A final clustering based on a subset of these boundaries is determined using different supervised and unsupervised algorithms. The proposed methods were evaluated experimentally and compared to existing approaches on two sets of test data. The automatic proposed methods' performance exceeds that of manually tuned alternatives in our testing, and have been well received by users of our photo management application.

ACKNOWLEDGMENTS

We wish to thank John Adcock for his work on the photo management application documented in [Girgensohn et al. 2003] that was used in the experimental evaluation. We thank Yong Rui for his readings of [Cooper et al. 2003] and the anonymous reviewers of this manuscript for their comments. Finally, we thank our colleagues at FX Palo Alto Laboratory and Fujifilm Software California for their feedback on the event clustering algorithm and its integration in our photo management application.

REFERENCES

Boreczky, J. and Rowe, L. 1996. Comparison of video shot boundary detection techniques. In *SPIE Storage and Retrieval for Image and Video Databases*. SPIE, Bellingham, WA, 170–179.

Cooper, M., Foote, J., Girgensohn, A., and Wilcox, L. 2003. Temporal event clustering for digital photo collections. In *Proceedings of the eleventh ACM international conference on Multimedia*. ACM, 364–373.

Duda, R. and Hart, P. 1973. *Pattern Classification and Scene Analysis*. Wiley-Interscience.

Foote, J. 2000. Automatic audio segmentation using a measure of audio novelty. In *Proceedings of the IEEE Intl. Conf. on Multimedia and Expo*. IEEE, 452–55.

Frohlich, D., Kuchinsky, A., Pering, C., Don, A., and Ariss, S. 2002. Requirements for photoware. In *Proceedings of the ACM Conf. on CSCW*. ACM, New York, 166–75.

Gargi, U. 2003. Modeling and clustering of photo capture streams. In *Proceedings of the 5th ACM SIGMM workshop on Multimedia information retrieval*. ACM, New York, 47–54.

Girgensohn, A., Adcock, J., Cooper, M., Foote, J., and Wilcox, L. 2003. Simplifying the management of large photo collections. In *Proceedings of Human-Computer Interaction INTERACT '03*. IOS Press, 196–203.

Graham, A., Garcia-Molina, H., Paepcke, A., and Winograd, T. 2002. Time as the essence for photo browsing through personal digital libraries. In *Proceedings of the Joint Conf. on Digital Libraries*. ACM, New York, 326–35.

Hartigan, J. 1975. *Clustering Algorithms*. Wiley & Sons.

Jaimes, A., Benitez, A. B., Chang, S.-F., and Loui, A. C. 2000. Discovering recurrent visual semantics in consumer photographs. In *IEEE Intl. Conf. on Image Processing, Vol. 2*. IEEE, 528–531.

JEIDA. 1998. *Digital Still Camera Image File Format Standard*. Japan Electronic Industry Development Association.

Kohonen, T. 1989. *Self-Organization and Associative Memory*. Springer-Verlag.

Kohonen, T., Kangas, J., Laaksonen, J., and Torkkola, K. 1992. Lvq pak: A program package for the correct application of learning vector quantization algorithms. In *Intl. Joint Conf. on Neural Networks*. ACM, New York, I 725–730.

Leung, Y., Zhang, J.-S., and Xu, Z.-B. 2000. Clustering by scale-space filtering. *IEEE Trans. on Pattern Analysis & Machine Intelligence 22,* 12, 1396–1410.

Lim, J.-H., Tian, Q., and Mulhelm, P. 2003. Home photo content modelling for personalized event-based retrieval. *IEEE Multimedia 10,* 4, 28–37.

Loui, A. and Savakis, A. 2003. Automatic event clustering and quality screening of comsumer pictures for digital albuming. *IEEE Trans. on Multimedia 5,* 3, 390–402.

Mills, T., Pye, D., Sinclair, D., and Wood, K. 2000. Shoebox: A digital photo management system. In *Technical Report 2000.10*. AT&T Laboratories Cambridge, Cambridge, UK.

Mojsilovic, A., Gomes, J., and Rogowitz, B. 2002. Isee: Perceptual features for image library navigation. In *SPIE Human Vision and Electronic Imaging*. SPIE, Bellingham, WA, 266–277.

Naaman, M., Song, Y. J., Paepcke, A., and Garcia-Molina, H. 2004. Automatic organization for digital photographs with geographic coordinates. In *Proceedings of the Joint Conf. on Digital Libraries*. ACM, New York, 326–35.

Platt, J., Czerwinski, M., and Field, B. 2003. Simplifying the management of large photo collections. In *Fourth IEEE Pacific Rim Conference on Multimedia*. IEEE, 6–10.

Rodden, K. 2002. Evaluating similarity-based visualisations as interfaces for image browsing. Ph.D. thesis, Univeristy of Cambridge.

Rodden, K. and Wood, K. 2003. How do people manage their digital photographs? In *Proceedings of the ACM Conf. on Human factors in computing systems (CHI)*. ACM, New York, 409–416.

Schwarz, G. 1978. Estimating the dimension of a model. *The Annals of Statistics 6*, 461–64.

Slaney, M., Ponceleon, D., and Kaufman, J. 2001. Multimedia edges: Finding hierarchy in all dimensions. In *ACM Intl. Conf. on Multimedia*. ACM, New York, 29–40.

Witkin, A. 1984. Scale-space filtering: A new approach to multi-scale description. In *IEEE ICASSP, Vol. 9*. IEEE, 150–3.

Received Someday; revised Someday later; accepted Someday even later than that