Temporal Event Clustering for Digital Photo Collections

Matthew Cooper, Jonathan Foote, Andreas Girgensohn, and Lynn Wilcox
FX Palo Alto Laboratory
3400 Hillview Ave. Bldg. 4
Palo Alto, CA USA
[cooper, foote, andreasg, wilcox]@fxpal.com

ABSTRACT

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 results for the algorithm based solely on temporal similarity, and jointly on temporal and content-based similarity. We also describe a supervised algorithm based on learning vector quantization. Finally, we include experimental results for the proposed algorithms and several competing approaches on two test collections.

Categories and Subject Descriptors

H.5 [**Information Interfaces and Presentation**]: Multimedia Information Systems; H.3 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*Indexing methods*

General Terms

algorithms, management

Keywords

digital photo organization, temporal media indexing and segmentation

1. INTRODUCTION

Digital cameras are coming into widespread use, and as a result, users are amassing increasingly large collections of digital photographs. There is thus a demand for automatic tools to help manage, organize, and browse these collections. A recent study focused on requirements for these tools [1]. The authors emphasized the importance of intuitive photo management software capable of supporting a variety of usage scenarios. Fortunately, digital photographs typically include metadata, such as the time and date, in a standard image header such as Exif (EXchangeable Image File [2]) that can be used for automatic organization. In the future, many consumer digital cameras will also record global positioning system (GPS) location information which should also prove valuable for this task.

Consumers often organize their photos in terms of "events" both for browsing and retrieval, as well as for sharing selected photos with others. Events are naturally associated with specific times and places, such as a child's birthday party or a vacation. However, events are difficult to define quantitatively or consistently. The photos associated with an event often exhibit little coherence in terms of both low-level image features and visual similarity. As an example of an event, consider the possible pictures taken during a trip to the beach. The photos could have widely different subjects such as the beach, the ocean, vehicles, or people.

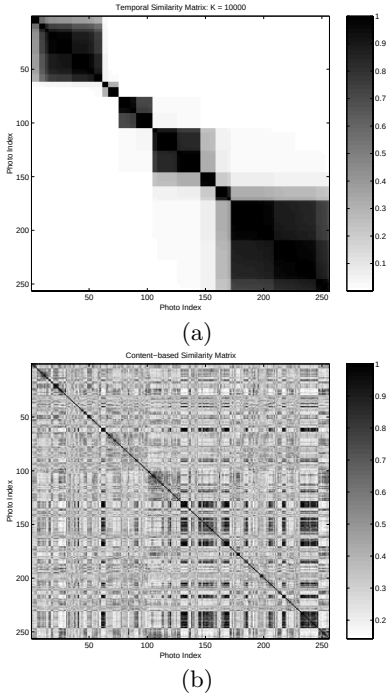

Figure 1: Panel (a) shows a temporal similarity matrix computed for 256 digital photos. Panel (b) shows the content-based similarity matrix calculated from low frequency DCT features and the cosine similarity measure.

MM'03, November 2–8, 2003, Berkeley, California, USA.

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. Stanford researchers recently reported that organizing photos by time significantly improves users' performance in a series of retrieval tasks [3].

To examine inter-photo similarity, we compute similarity matrices for 256 photos using both temporal and content based features in Figure 1. The 256 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 depicted in Figure 2. Specifically, the (i, j) element of the matrix quantifies the similarity between the i^{th} and j^{th} photos. Throughout, photos are ordered according to their timestamps. Figure 1(a) shows the temporal similarity matrix computed as

$$\mathbf{S}(i,j) = \exp\left(-\frac{|t_i - t_j|}{10000}\right)$$

where t_i and t_j are the timestamps in minutes of photos i and j, respectively. The blocks of high similarity along the main diagonal of the matrix indicate groups of photos with similar timestamps. A checkerboard pattern along the main diagonal indicates the boundary between two such groups. The crux of the checkerboard pattern is the boundary in time order between the photos in the two events. The matrix provides a reasonably clear visualization of the temporal structure of the photos. Large blocks of high similarity appear along the main diagonal of the similarity matrix. Figure 1(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:

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

$\mathbf{v}_i$ denotes the DCT features of the i^{th} photo in time order. Far less structure is evident in $\mathbf{S}_C$, compared to the temporal similarity matrix of panel (a). In our experience, content-based image similarity is less useful for photo clustering and event detection than metadata.

We focus here on analyzing the photos' timestamps, though our goal is a general framework in which metadata and content-based information are integrated for automatic photo organization. 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.

Our approach is based on studying the similarity between the photos' timestamps. The first step is to extract and sort the timestamps in a photo collection. We quantify temporal similarity by pairwise comparisons of timestamps in local neighborhoods moving through the collection. We have adapted a media segmentation algorithm [4] to calculate a photo-indexed novelty score. As detailed in Section 3, the novelty score measures the intra-class similarity and inter-class dissimilarity between adjacent groups of photos. Specifically, the novelty score quantifies the similarity of the groups of photos taken both before and after a candidate boundary in time order. We assume that the photos at event

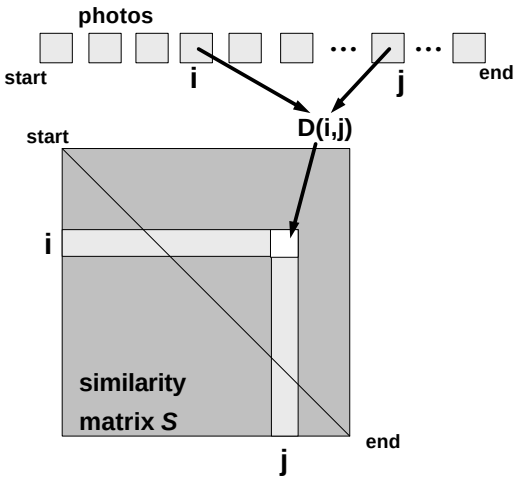

Figure 2: Embedding photo similarity data in a matrix.

boundaries separate two adjacent groups of photos with high intra-class temporal similarity and low inter-class similarity. 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. Using these multi-scale features, we present a supervised algorithm for event detection. We also train a learning vector quantizer (LVQ) to classify each photo's features as either an event "boundary" or "interior".

Next, we detail 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. Additionally, we present a version of this algorithm which integrates content-based and temporal similarity using a simple heuristic.

We reported preliminary results for unsupervised event detection in [5]. The algorithm has now been implemented in an application for organizing digital photos [6]. In this paper, we introduce approaches based on scale-space analysis and learning vector quantization and expand the experimental evaluation. Our unsupervised similarity-based approach is fully automatic and its performance approximates that of hand-tuned semi-automatic techniques (i.e. algorithms with thresholds that are manually set to maximize performance).

The analytic 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.

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. Sections 4 and 5 detail the supervised and unsupervised algorithms for event detection. In Section 6, we present ex-

perimental results comparing the proposed approaches and competing methods on two test collections of digital photos classified into meaningful events by the photographer. The paper concludes with a summary discussion.

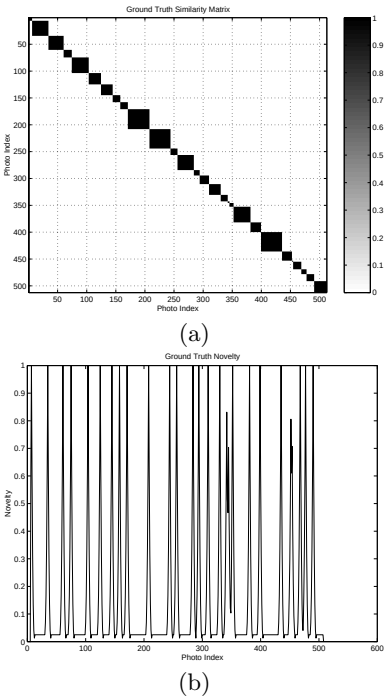

Figure 3: Panel (a) shows the ground truth similarity matrix. Panel (b) shows the novelty score computed using a gaussian checkerboard kernel.

2. RELATED WORK

Automatic digital photo organization has received increased attention in recent years. The algorithms in [3, 7] 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 [8]. 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 [9]. In [10], semantically-motivated content-based features were developed for image indexing and retrieval without the use of metadata.

Our work is closer in spirit to scale-space analysis [11, 12] and its application to the segmentation of text and video streams in [13]. 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. We present a general framework in which provably useful semantic or other content-based features may be integrated in future work. Our approach, detailed below, is fully automatic and requires no thresholds or training. Unlike [3, 7], temporal similarity is assessed at multiple scales, and the similarity measure is calculated between *all pairs* of points in local 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 use a confidence measure that determines a "best" scale for the event segmentation over the entire collection of photos. Unlike [13], 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 confidence measure of Section 5.2 to compare clustering performance at the different scales. 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, $\{t_i : i = 1, \cdots, N\}$, 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 the analysis in [4], because the time difference between indices (photos) is non-uniform. Thus, each photo is represented by its scalar timestamp.

3.1 Distance matrix embedding

Our approach is founded on similarity analysis. As an example, Figure 3(a) shows the similarity matrix generated from the ground truth clustering of 500 photos from our test set. Each photo in the test set 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 photos are indexed in time order, and the (i, j) element of the matrix compares the names of the folders in which i^{th} and j^{th} photos were stored. This embedding is graphically depicted in Figure 2. The 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 crux of the checkerboard pattern is the boundary between the photos in the two events.

This is a convenient visualization, which 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.

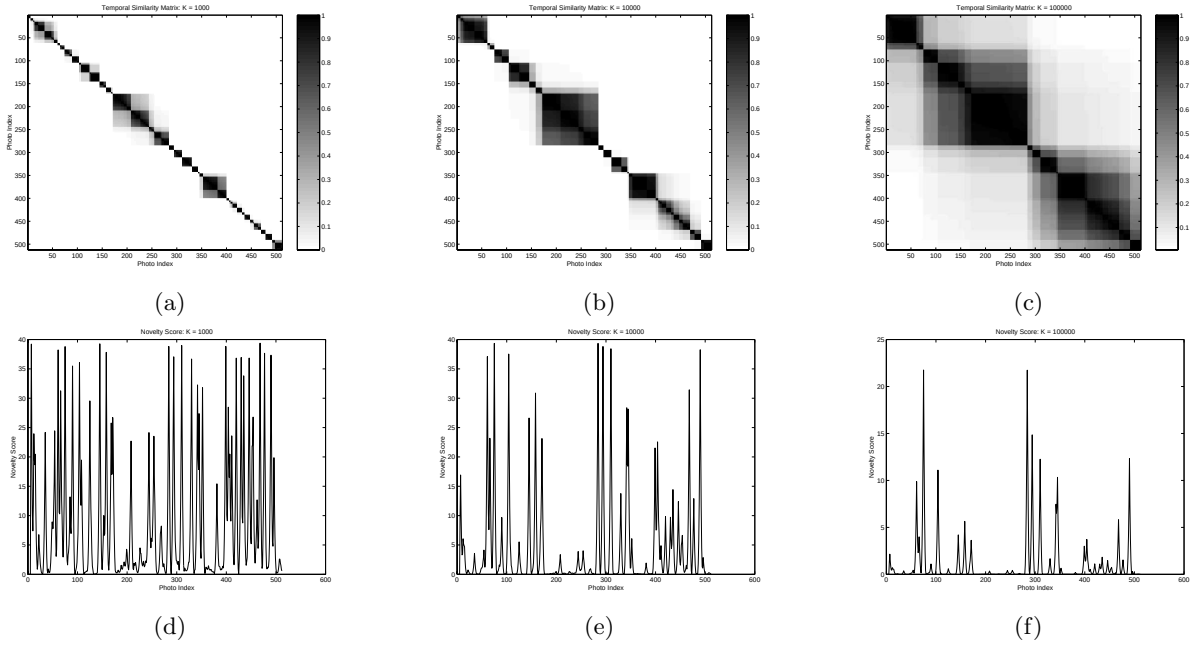

Figure 4: The left column shows the similarity matrices $\mathbf{S}_K$ for $K = 1000$ (a), $K = 10000$ (b), and $K = 100000$ (c) minutes. Panels (d), (e), and (f) show the corresponding novelty scores computed using a Gaussian checkerboard kernel.

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 exponential similarity measure. For calculation, the units of K and the timestamps are minutes; the similarity measure is unitless. By varying K, we can visualize the similarity between the timestamps at differing granularities. The top row of Figure 4 shows similarity matrices computed using (2) for $K = 10^3, 10^4, 10^5$ 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 3(a), the event clusters are visible as dark 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 [4]. We seek the centers of the checkerboards; each corresponds to the boundary between two adjacent groups of photos each with high temporal *self-similarity*. The off-diagonal squares of the checkerboard indicate low temporal *cross-similarity*. The novelty score quantifies local self-similarity and cross-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

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

An example kernel appears in Figure 5. Throughout, $\ell = 6$, so that the kernel is 12×12. The bottom row of Fig. 4 shows the novelty scores computed for $K = 10^3, 10^4, 10^5$ 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.

Figures 3 and 4 show that the matrices are typically zero 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 the same width as the kernel, reducing computational complexity to order N. We use the set of novelty scores computed using matrices with varying values for K as features for photo event clustering. Let K take M values in the analysis: $K \in \mathcal{K} \equiv \{K_1, \cdots, K_M\}$. Then, the total algorithmic complexity is order $N \cdot M$. In the following sections, we present supervised and unsupervised algorithms for event clustering based on multi-scale analysis of local temporal similarity.

4. SUPERVISED EVENT CLUSTERING

In this section, we describe a supervised algorithm for event clustering based on a LVQ [14]. Here, we assume that the novelty features can be used to distinguish photos at event boundaries from the remainder of the collection. Define the $M \times N$ matrix $\mathcal{N}(j,i) = \nu_{K_j}(i)$. We associate

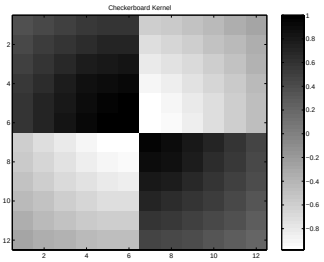

Figure 5: An example checkerboard kernel used to compute the novelty features.

photo i with its novelty scores at each scale,

$$\mathcal{N}_i \equiv \begin{bmatrix} \mathcal{N}(K_1, i) \\ \vdots \\ \mathcal{N}(K_M, i) \end{bmatrix} = \begin{bmatrix} \nu_{K_1}(i) \\ \vdots \\ \nu_{K_M}(i) \end{bmatrix} \quad . \tag{4}$$

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 not correspond to maxima in the novelty scores (at any scale). The LVQ is designed to provide effective event detection by discriminating between these classes.

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,

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

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 [15]. The codebook vectors for each class are used for nearest-neighbor classification [16] of the novelty features for each photo in the test set.

While the approach is non-parametric and discriminative, the 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 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 this 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 training data for each scale $K \in \mathcal{K}$:*
   (a) *Compute the similarity matrix $\mathbf{S}_K$ using* (2).
   (b) *Compute the novelty score ν_K of* (3).
2. *Train LVQ using the iterative procedure of* (5). *This can be done off-line.*
3. *Calculate novelty features for the testing data for each $K \in \mathcal{K}$*
   (a) *Compute the similarity matrix $\mathbf{S}_K$ using Eq.* (2).
   (b) *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

In this Section, we present three unsupervised algorithms for 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 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 [11, 12]. 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 a set of signals

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

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

where $2L+1$ 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.

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*
   (a) *Compute* T_σ *as in* (7).
   (b) *Detect peaks in* T_σ*, tracing peaks from larger scales to smaller scales.*

We do not include a final step for peak selection. In practice, various criteria are used to select the peaks in scale space that comprise the final event segmentation (e.g. see [12]). We use $L = 10$ in the experiments of Section 6.

5.2 Time-based similarity analysis

The similarity-based approach processes the novelty measures of (3). We first locate peaks at each scale $K \in \mathcal{K}$ 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.

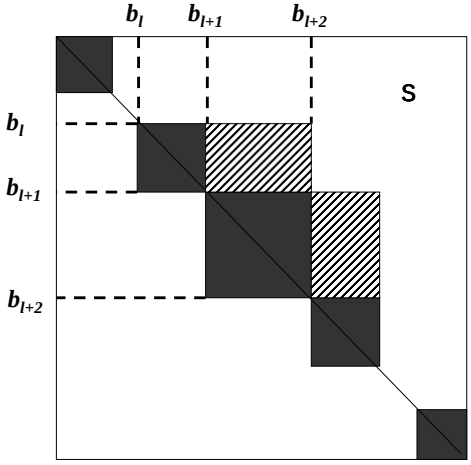

Figure 6: Computing a confidence score for clustering. The dark regions represent within-cluster similarity, while the gray regions represent between-cluster similarity.

This procedure results in a list of cluster boundaries and strengths at multiple resolutions. In traditional scale-space analysis, the number of scales over which a peak can be traced has been used to assess its importance as a boundary. Our current approach is to select a single, best, resolution level. To determine the "goodness" of the boundaries at a given time scale, we calculate a confidence measure from the average within-class similarity and the between-class 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

$$C(\mathcal{B}_K) = \sum_{l=1}^{|\mathcal{B}_K|-1} \sum_{i,j=b_l}^{b_{l+1}} \frac{\mathbf{S}_K(i,j)}{(b_{l+1}-b_l)^2} - \sum_{l=1}^{|\mathcal{B}_K|-2} \sum_{i=b_l}^{b_{l+1}} \sum_{j=b_{l+1}}^{b_{l+2}} \frac{\mathbf{S}_K(i,j)}{(b_{l+1}-b_l)(b_{l+2}-b_{l+1})} \quad . \tag{8}$$

The first term above quantifies the average within-class similarity between the photos within each cluster. The second term quantifies the average between-class similarity between photos in adjacent clusters. By negating this term, the confidence measure thus combines each cluster's average self-similarity and the dissimilarity between adjacent clusters. Fig. 6 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. Algorithm 3 details the computational steps.

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

1. *Extract and sort photo timestamps,* $\{t_1, \cdots, t_n\}$.
2. *For each* K *in decreasing order*
   (a) *Compute the similarity matrix* $\mathbf{S}_K$ *using Eq.* (2).
   (b) *Compute the novelty score* ν_K *of Eq.* (3).
   (c) *Detect peaks in* ν_K.
   (d) *Form event boundary list using event boundaries from previous iterations and newly detected peaks.*
3. *Compute confidence score using list of event boundaries,* $\mathcal{B}_K$ *for each* K *following Eq.* (8).
4. *Select event boundary list for* K *maximizing the confidence score.*

5.3 Time and content-based analysis

We have 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):

$$\mathbf{S}_K^{(J)}(i,j) = \begin{cases} \mathbf{S}_K(i,j) & \text{if } |t_i - t_j| > 48 \text{ hours} \\ \alpha \mathbf{S}_K(i,j) + (1-\alpha)\mathbf{S}_C(i,j) & \text{otherwise.} \end{cases} \tag{9}$$

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

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

$$\mathbf{S}_K^{(J)}(i,j) = \begin{cases} \mathbf{S}_K(i,j) & \text{if } |t_i - t_j| > 48 \text{ hours} \\ \max(\mathbf{S}_C(i,j), \mathbf{S}_K(i,j)) & \text{otherwise.} \end{cases} \tag{10}$$

Table 1: The algorithms used in our experiments. The second column indicates whether the algorithm is supervised (Sup.) or unsupervised (Unsup.).

Algorithm	Sup.	Application	Description
Adaptive Threshold 1	Unsup.	Hand-tuned	adaptive threshold of inter-photo time difference: see [7]
Adaptive Threshold 2	Unsup.	Hand-tuned	adaptive threshold of inter-photo time difference: see [3]
Threshold	Unsup.	Hand-tuned	fixed threshold of inter-photo time difference
Scale-space	Unsup.	Hand-tuned	fixed threshold of peaks in scale-space: see Section 5.1
LVQ	Sup.	Automatic	nearest-neighbor classifier: see Section 4
Temporal Similarity	Unsup.	Automatic	automatic peak detection from novelty scores: see Alg 3
Joint Similarity	Unsup.	Automatic	automatic peak detection from novelty scores: see Section 5.3

The heuristic used to build $\mathbf{S}_K^{(J)}$ 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 (10) has consistently outperformed that of (9), and we use (10) 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.

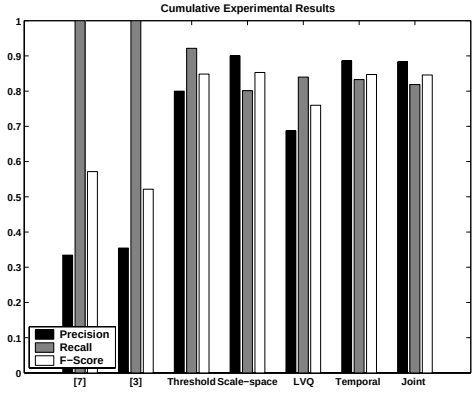

Figure 7: Experimental comparison of several algorithms for photo event clustering.

5.4 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 $O(N^2)$. In practice, we pre-compute only the portion of the pairwise distance matrix along the main diagonal with width ℓ as in (3), and next evaluate (2) for each scale. Peaks are detected in each of the N-dimensional novelty scores by computing first differences. The complexity of the evaluation of the confidence score is more difficult. It is this step that potentially necessitates the computation of the entire lookup table, since the extent of events can't be assumed in advance (otherwise, the lookup table could be limited to the same strip around the main diagonal of width $2 \cdot \ell = 12$). In the worst case, the sums of (8) include all N^2 terms of $\mathbf{S}_K$. Thus, given the lookup table of inter-photo time differences, the computation of the M-dimensional feature vectors of (4) is $O(N \cdot M)$. Because the temporal similarity measure decays exponentially as the time difference increases, we can also 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 have not found these speedups to be necessary. In practice, we provide a fully automatic solution by using the confidence measure to select a single scale for the detected event boundaries. To quantify this point we include representative runtimes for the temporal-version of Algorithm 3 on a collection of 3931 photos in Table 3. The column labelled "No Conf." is the time for step 2 in the algorithm. The column labelled "Conf." is the time for the entire algorithm. The event detector has been implemented in java as part of the application documented in [6], and the times here were produced using a PC with a 2.4 GHz Pentium 4 processor. As predicted, after doubling the number of photos processed (N), the time for the segmentation step increases linearly, while including the confidence measure incurs an exponential cost. Nonetheless, the overall runtime is fast, even for a reasonably large number of photos. In practice, content-based processing, such as thumbnail extraction, is more computationally expensive than event detection, and for temporal similarity, we process only a single scalar feature per image.

Table 3: The tables documents run times for a typical photo collection. The times are in seconds.

Run times (3931 photos total)		
N	No Conf.	Conf.
983	0.062	0.9125
1966	0.086	2.818
3931	0.164	8.152

[1]Using (9), the results for Collection I of Table 2 are precision = 0.8, recall = 0.62, F-score = 0.7. For Collection II, precision = 0.74, recall = 0.78, F-score = 0.76.

Table 2: The two tables summarize our experimental results.

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 Similarity	0.884	0.807	0.83
Joint Similarity	0.9	0.79	0.84

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 Similarity	0.89	0.89	0.89
Joint Similarity	0.84	0.89	0.86

6. EXPERIMENTAL RESULTS

In the previous Sections, we reviewed and presented several algorithms for event detection. Here, we compare the event clustering performance of seven algorithms 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 1 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 13).

"Adaptive Threshold 1" is based on [7] and "Adaptive Threshold 2" is based on [3]. 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. 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 of photos for testing. For each test set, we train an LVQ using the remaining data and the its ground truth labelling. The results of the three separate tests are combined for comparison with the remaining unsupervised approaches.

The precision, recall, and F-score for the detected event boundaries appear in Table 2 for each algorithm. These measures are common figures of merit in information retrieval that are also used to assess segmentation performance [17]. Precision indicates the proportion of falsely labelled boundaries (over-segmentation):

$$\text{precision} = \frac{\text{correctly detected boundaries}}{\text{total number of detected boundaries}} \quad . \quad (11)$$

Recall measures the proportion of true boundaries detected:

$$\text{recall} = \frac{\text{correctly detected boundaries}}{\text{total number of ground truth boundaries}} \quad . \quad (12)$$

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

$$\text{F-score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad . \quad (13)$$

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 both *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 7. In that

graph, [7] and [3] corresponds to the algorithms Adaptive Threshold 1 and Adaptive Threshold 2, respectively.

Figure 8: 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.

7. SUMMARY

We have presented several approaches to automatic event clustering for digital photo collections. The basic framework is to first quantitatively assess structure in the collection at multiple scales, and then feed this data into several different classifiers. Supervised and unsupervised algorithms were developed and presented, and compared to existing approaches on two sets of test data. We intend to improve our supervised algorithm using robust statistical techniques to mitigate the impact of outliers in the training data. We are also developing approaches to building event boundary lists directly the hierarchical boundary tree, as alternatives tot he confidence measure of (8). Such an approach will accommodate variability in event duration through a large photo collection.

In practice, we employ the automatic temporal similarity-based method (Algorithm 3 of Section 5.2). It has been well received by the pilot users of our application for organizing digital photos [6]. 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 8 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 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. To change photos' event membership, users simply drag and drop thumbnails onto the desired event label.

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 such as GPS information. Here, we included an initial attempt at combining metadata and content-based features in (10). 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.

8. REFERENCES

[1] D. Frohlich, A. Kuchinsky, *et al.*. Requirements for Photoware. *Proc. ACM CSCW*, pp. 166-75, 2002.

[2] Digital Still Camera Image File Format Standard. Japan Electronic Industry Development Association, 1998. http://www.pima.net/standards/it10/PIMA15740/exif.htm

[3] A. Graham, H. Garcia-Molina, A. Paepcke, and T. Winograd. Time as the essence for Photo Browsing Through Personal Digital Libraries. *Proc. Joint Conf. on Digital Libraries*, 2002.

[4] J. Foote, Automatic Audio Segmentation using a Measure of Audio Novelty. *Proc. IEEE Intl. Conf. on Multimedia and Expo*, 2000.

[5] M. Cooper, J. Foote, and A. Girgensohn. Automatically Organizing Digital Photographs Using Time and Content. To appear *Proc IEEE ICIP*, 2003.

[6] A. Girgensohn, J. Adcock, M. Cooper, J. Foote, and L. Wilcox. Simplifying the Management of Large Photo Collections. *Human-Computer Interaction INTERACT '03*, IOS Press, 2003 (to appear).

[7] J. Platt, M. Czerwinski, and B. Field. PhotoTOC:Automatic Clustering for Browsing Personal Photographs. Microsoft Research Techincal Report MSR-TR-2002-17, 2002.

[8] A. Loui and A. Savakis. Automatic Image Event Segmentation and Quality Screening for Albuming Applications. *Proc. IEEE Intl. Conf. on Multimedia and Expo*, 2000.

[9] A. Jaimes, A. B. Benitez, S.-F. Chang, and A. C. Loui. Discovering Recurrent Visual Semantics in Consumer Photographs. *Proc. IEEE Intl. Conf. on Image Processing*, 2000.

[10] A. Mojsilovic, J. Gomes, and B. Rogowitz. ISee: Perceptual Features for Image Library Navigation. *Proc. SPIE Human Vision and Electronic Imaging*, 2002.

[11] A. Witkin. Scale-space Filtering: A New Approach to Multi-scale Description. *Proc. IEEE ICASSP*, 1984.

[12] Y. Leung, J.-S. Zhang, and Z.-B. Xu. Clustering by Scale-space Filtering. *IEEE Trans. on Pattern Analysis & Machine Intelligence*, **22**(12):1396-1410, 2000.

[13] M. Slaney, D. Ponceleon, and J. Kaufman. Multimedia Edges: Finding Hierarchy in all Dimensions. *Proc. ACM Conf. on Multimedia*, 2001.

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

[15] T. Kohonen and J. Kangas and J. Laaksonen and K. Torkkola. LVQ PAK: A program package for the correct application of Learning Vector Quantization algorithms. *Proc. of the Intl. Joint Conf. on Neural Networks*, pp. I 725–730, 1992.

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

[17] J. Boreczky and L. Rowe. Comparison of video shot boundary detection techniques. *Proc. SPIE Storage and Retrieval for Image and Video Databases*, 1996.