Audio-Visual Atoms for Generic Video Concept Classification

WEI JIANG, COURTENAY COTTON, SHIH-FU CHANG, DAN ELLIS, Columbia University
ALEXANDER C. LOUI, Eastman Kodak Company

We investigate the challenging issue of joint audio-visual analysis of generic videos targeting at concept detection. We extract a novel local representation, Audio-Visual Atom (AVA), which is defined as a region track associated with regional visual features and audio onset features. We develop a hierarchical algorithm to extract visual atoms from generic videos, and locate energy onsets from the corresponding soundtrack by time-frequency analysis. Audio atoms are extracted around energy onsets. Visual and audio atoms form AVAs, based on which discriminative audio-visual codebooks are constructed for concept detection. Experiments over Kodak's consumer benchmark videos confirm the effectiveness of our approach.

Categories and Subject Descriptors: H.3.1 [**Information Storage and Retrieval**] Content Analysis and Indexing; H.4.m [**Information Systems Applications**] Miscellaneous; H.3.m [**Information Storage and Retrieval**] Miscellaneous

General Terms: Algorithms, Experimentation

Additional Key Words and Phrases: Semantic concept detection, joint audio-visual analysis, audio-visual atom, audio-visual codebook

ACM Reference Format:

Jiang, W., Cotton, C., Chang, S.-F., Ellis, D., and Loui, A. C. 2010. Audio-visual atoms for generic video concept classification. ACM Trans. Multimedia Comput. Commun. Appl. 6, 3, Article 14 (August 2010), 19 pages.
DOI = 10.1145/1823746.1823748 http://doi.acm.org/10.1145/1823746.1823748

1. INTRODUCTION

This article investigates the problem of automatic detection of semantic concepts in unconstrained videos, by joint analysis of audio and visual content. These concepts include generic categories, such as scene (e.g., beach), event (e.g., birthday, wedding), location (e.g., museum, playground) and object (e.g., animal, boat). Unconstrained videos are captured in an unrestricted manner, like those videos taken by consumers and uploaded to YouTube. This is a difficult problem due to the diverse video content as well as the challenging condition such as uneven lighting, clutter, occlusions, and complicated motions of both objects and camera.

This work has been supported by a gift from Eastman Kodak, NSF grants CNS-07-16293 and CNS-07-51078, and the New York State NYSTAR CATT program. W. Jiang has also been supported as a Kodak Graduate Fellow.

Authors' addresses: W. Jiang, C. Cotton, S.-F. Chang, and D. Ellis, Electrical Engineering Department, Columbia University, New York, NY 10027; W. Jiang (corresponding author): email: wjiang@ee.columbia.edu; A. C. Loui, Kodak Research Labs, Eastman Kodak Company, Rochester, NY 10027.

DOI 10.1145/1823746.1823748 http://doi.acm.org/10.1145/1823746.1823748

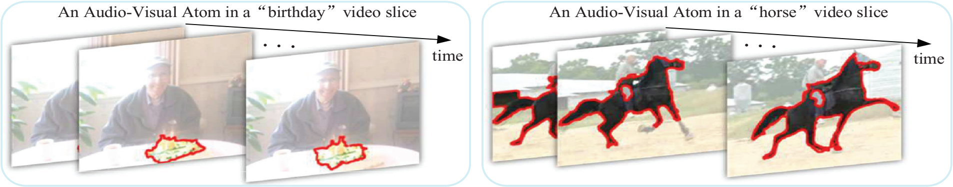
An Audio-Visual Atom in a “birthday” video slice
time
An Audio-Visual Atom in a “horse” video slice
time

Fig. 1. Examples of audio-visual atoms. The region track of a birthday cake and the background birthday music form a salient audio-visual cue to describe “birthday” videos. The region track of a horse and the background horse running footstep sound give a salient audio-visual cue for the “horse” concept.

To exploit the power of both visual and audio aspects for video concept detection, *multimodal fusion* approaches have attracted much interest [Anemueller et al. 2008; Chang et al. 2007]. Visual features such as color and texture are extracted from the whole images at the global level, and audio features such as MFCCs are generated from the audio signal in the same time window. In early fusion methods [Chang et al. 2007], such audio and visual raw features at the global level are either directly fused by concatenation to learn classifiers or used to generate individual kernels which are then added up into a fused kernel for classification. In late fusion approaches [Anemueller et al. 2008; Chang et al. 2007], detectors are first trained over audio and visual features respectively and then fused to generate the final detection results. These fusion methods have shown promising performance improvements. However, visual features over the whole frame are insufficient to capture the object information, and the disjoint process of extracting audio and visual features limits the ability to generate joint audio-visual patterns that are useful for concept detection. For example (in Figure 1), the joint pattern of a birthday cake region and the birthday music is an intuitive strong audio-visual cue at the object level for the “birthday” concept but has never been explored in prior works that are primarily based on global features or global level fusion.

On the other hand, there are many recent works exploring audio-visual analysis for object detection and tracking. In audio-visual speech and speaker recognition [Iwano et al. 2007; Kaucic et al. 1996], visual features obtained by tracking the movement of lips, mouths, and faces are combined with audio features describing acoustic speech for improved classification. By using multiple microphones, the spatial location of sounding sources can be estimated by stereo triangulation, and can be combined with object motion captured by one or multiple cameras to enhance tracking [Beal et al. 2003]. For audio-visual object localization in a single input containing one video stream associated with one sound track, the audio-visual synchrony in time is used to study audio-visual correlation [Barzelay and Schechner 2007; Cristani et al. 2007]. For example, under the assumption that fast moving pixels make big sounds, temporal patterns of significant changes in the audio and visual signals are found and their correlation is maximized to locate sounding pixels [Barzelay and Schechner 2007]. Such audio-visual object detection and tracking methods have shown interesting results in analyzing videos in a controlled environment where good foreground/background separation can be obtained, for example, in surveillance applications. However, both object detection and tracking are known to be extremely difficult in generic videos. There usually exist uneven lighting, clutter, occlusions, and complicated motions of both objects and camera. Also, unconstrained sound tracks are generally generated by multiple sound sources under severe noises. Blind sound source separation in such scenarios remains challenging. In addition, the audio-visual synchronization cannot be observed most of the time. Multiple objects usually make sounds together, with or without large movements, and often some objects making sounds do not appear in the video.

In this work, we investigate the challenging issue of audio-visual analysis in generic videos aiming at detecting generic concepts. We propose a novel multimodal representation, Audio-Visual Atom (AVA), by extracting atomic representations from both visual and audio signals of the video. We track automatically segmented regions based on the visual appearance within a short video slice (e.g., 1 sec). Regional visual features (e.g., color, texture, and motion) can be extracted from such short-term region tracks. Then based on the visual similarity short-term region tracks from adjacent short-term video slices are connected into long-term region tracks that are called visual atoms. At the same time we locate audio energy onsets from the corresponding audio soundtrack by decomposing the audio signal into the most prominent bases from a time-frequency representation. A short-term audio signal is reconstructed within a short window around each local energy onset, which is called an audio atom. Audio features (e.g., spectrogram) can be generated to describe audio atoms. Then visual atoms and audio atoms are combined to form a joint audio-visual atomic representation, that is, AVAs. Based on AVAs, joint audio-visual codebooks can be constructed, and the codebook-based features can be used for concept detection.

Our method provides a balanced choice for exploring audio-visual correlation in generic videos: compared to the previous audio-visual fusion approach using coarsely aligned concatenation of global features [Chang et al. 2007], we generate an atomic representation in which a moderate level of synchronization is enforced between region tracks and local audio onsets; compared to the tight audio-visual synchronization framework focusing on object detection and tracking [Cristani et al. 2007], we do not rely on precise object extraction. Compared to alternative methods using static image frames without temporal tracking, less noisy atomic patterns can be found by the short-term tracking characteristics of AVAs. As illustrated by the AVA examples in Figure 1, the temporal region track of a birthday cake associated with the background birthday music gives a representative audio-visual atomic cue for describing “birthday” videos. Similarly, the temporal horse region track together with the horse running footstep sound form a joint audio-visual atomic cue that is salient for describing the “horse” concept. Figure 1 also indicates that the joint audio-visual correlation captured by our AVA is based on co-occurrence, for example, frequent co-occurrence between a birthday cake and the birthday music. Accordingly, the audio-visual codebooks constructed from salient AVAs can capture the representative audio-visual patterns to describe different individual concepts, and significant detection performance improvements can be achieved.

2. OVERVIEW OF OUR APPROACH

Figure 2 shows the framework of our system. The audio and visual processes of our method share commonality in their conceptual approaches to decomposing signals into short-term atomic entities, instead of requiring tight synchronization or long-term tracking. The emphasis on short-term structures is manifested in the visual part by performing visual region tracking over very short time slices (e.g., 1 second), and in the audio part by detecting transient audio phenomena associated with peak energies with common onset in frequency and time. Our main objective then is to discover the statistical relations between such short-term co-occurring atoms across modalities. We will briefly summarize our work in this section. More details about the visual and audio processes can be found in Section 3 and Section 4, respectively.

In the visual aspect, we propose a hierarchical framework to extract visual atoms from unconstrained videos. In the first step, we develop an effective algorithm, named Short-Term Region tracking with joint Point Tracking and Region Segmentation (STR-PTRS), to extract short-term region tracks within short windows. STR-PTRS accommodates the challenging conditions in unconstrained videos by: conducting temporal tracking within short-term video slices (e.g., 1 sec); spatially localizing

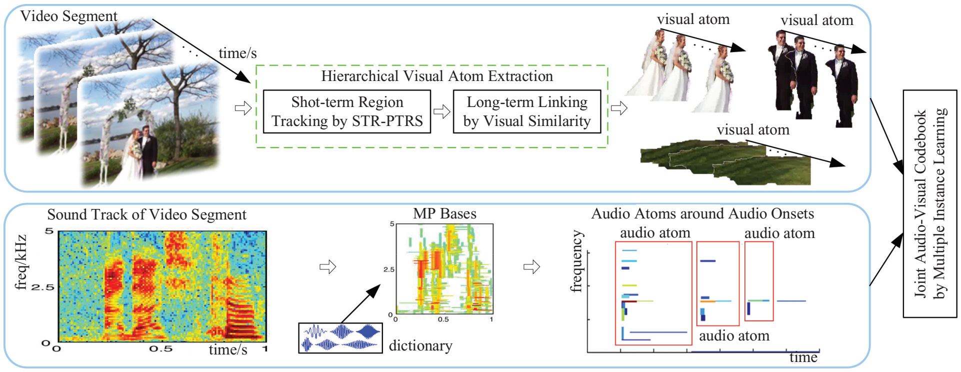
Video Segment
time/s
Hierarchical Visual Atom Extraction
Shot-term Region Tracking by STR-PTRS
Long-term Linking by Visual Similarity
visual atom
visual atom
visual atom
Joint Audio-Visual Codebook by Multiple Instance Learning
Sound Track of Video Segment
freq/kHz
5
2.5
0
0
0.5
time/s
1
MP Bases
dictionary
Audio Atoms around Audio Onsets
audio atom
audio atom
audio atom
frequency
time

Fig. 2. The overall framework of the proposed joint audio-visual analysis approach.

meaningful regions by image segmentation based on color and texture; and jointly using interest point tracking and region segmentation. The short-term region tracks are not restricted to foreground objects. They can be foreground objects or backgrounds, or combinations of both, all of which carry useful information for detecting various concepts. For example, the red carpet alone or together with the background wedding music are important for classifying the "wedding" concept. Visual features such as color, texture, and spatial location can be generated from short-term region tracks. Then in the second step, we connect short-term region tracks from adjacent short-term video slices into long-term visual atoms according to visual similarities calculated using visual features.

With temporal tracking in short-term video slices, better visual atomic patterns can be found compared to the static-region-based alternatives where no temporal tracking is involved. Tracking of robust regions can reduce the influence of noisy regions. Such noise usually comes from imperfect segmentation, for example, over segmentation or wrong segments due to sudden changes of motion or illumination. By finding trackable visual regions and using such region tracks as whole units to form the final visual atoms, the influence of erroneous segments from a few frames can be alleviated through averaging across the good segments as majorities.

The audio descriptors are based on a Matching Pursuit (MP) representation. MP [Mallat and Zhang 1993] is an algorithm for sparse signal decomposition from an over-complete set of basis functions, and MP-based audio features have been used successfully for classifying ambient environmental sounds [Chu and Narayanan 2008]. MP basis functions correspond to concentrated bursts of energy localized in time and frequency and span a range of time-frequency trade-offs, allowing us to describe an audio signal with the basis functions that most efficiently explain its structure. The sparseness of the representation makes this approach robust to background noise, since a particular element will remain largely unchanged even as the surrounding noise level increases; the representation is analogous to the one used in Ogle and Ellis [2007] in which energy peaks in time-frequency were used to remove framing issues and achieve robust audio matching against background noises. The composition of an MP representation should allow discrimination among the various types of structured (e.g., speech and music) and unstructured audio elements that are relevant to concept detection. We extract the audio atoms from the audio soundtrack corresponding to the video window where visual atoms are tracked. Each video window is decomposed into its most prominent elements, and energy onsets are located from the audio signal. Audio atoms are generated as the reconstructed audio sounds within the short windows around energy onsets and are described by the spectrogram features associated with the short windows.

Visual atoms and audio atoms from the same video window are associated with each other to generate AVAs. Based on the AVA representation, we construct discriminative audio-visual codebooks using the Multiple Instance Learning (MIL) technique [Maron and Lozano-Pérez 1998] to capture the representative joint audio-visual patterns that are salient for detecting individual concepts. Precise object detection and tracking are helpful but not required for our approach. This enables us to conduct audio-visual analysis in unconstrained videos, different from the previous methods like Cristani et al. [2007] focusing on audio-visual object detection and tracking. We extensively evaluate our algorithm over the challenging Kodak's consumer benchmark video set from real users [Loui et al. 2007]. Our method is compared with two state-of-the-art static-region-based image categorization approaches that also use MIL: the DD-SVM algorithm [Chen and Wang 2004], where visual codebooks are constructed by MIL based on static regions and codebook-based features are used for SVM classification; and the ASVM-MIL algorithm [Yang et al. 2006], where asymmetrical SVMs are directly built using static regions under the MIL setting. Experiments demonstrate significant improvements achieved by our joint audio-visual codebooks, for example, over 120% MAP gain (on a relative basis) compared to both DD-SVM and ASVM-MIL. In addition, the joint audio-visual features outperform visual features alone by an average of 8.5% (in terms of AP) over 21 concepts, with many concepts achieving more than 20%.

3. VISUAL ATOM EXTRACTION

Detecting and tracking unconstrained objects in generic videos are known to be difficult. There exist dramatic clutter, occlusions, change of shape and angle, and motions of both camera and objects. Most previous tracking algorithms, both blob-based trackers [Stauffer and Grimson 2002] and model-based trackers [Jepson et al. 2003] can not work well. Specifically, blob-based approaches rely on silhouettes derived from variants of background substraction methods, while in generic videos due to the complex motions from both objects and camera, and the occlusion of objects, it is very hard to obtain satisfactory silhouettes. Most model-based algorithms rely on manual initialization that is not available for automatic concept detection. Object detectors can be used to initialize a tracking process [Niebles et al. 2008] but are restricted to tracking some specific objects like human body or vehicle, since it is unrealistic to train a detector for any arbitrary object.

We propose an effective hierarchical framework to extract visual atoms from unconstrained videos. In the first step, we temporally track consistent short-term visual regions by an STR-PTRS method. Tracking is conducted within short-term video slices (e.g., 1 sec) to accommodate unconstrained videos. Only during a short period of time, the changes and movements of the camera and objects are relatively small and there is a high chance to find consistent parts in the frames that can be tracked well. To obtain meaningful regions from a video slice, we use the image segmentation algorithm (that relies on the static color and texture appearance) instead of the background substraction or spatial-temporal segmentation methods [Dementhon and Doermann 2003; Galmar and Huet 2007] that rely on motion. This is because it is very hard to separate camera motion from object motion in unconstrained videos and the overall motion is unstable. In addition, for semantic concept detection not only foreground objects but also backgrounds are useful.

Within each short-term video slice, we jointly use interest point tracking and region segmentation to obtain short-term region tracks. Robust points that can be locked-on well are tracked through the short-term video slice, and based on point linking trajectories, image regions from adjacent frames are connected to generate region tracks. Compared to other possible alternatives, for example, connecting regions with the similarity over the color and texture appearance directly, our approach is more effective in both speed and accuracy: to track a foreground/background region, matching with raw pixel values is not as reliable as matching with robust interest points, due to the change of lighting, shape, and angle; and extracting higher-level color or texture visual features for region matching is quite slow.

The next subsection will describe STR-PTRS in detail. Now let's formulate our problem. Let $\mathbf{v}$ denote a video that is partitioned into video segments $u_1, \ldots, u_L$ (with fixed-length intervals or shot segmentation boundaries). Each video segment u (we omit index l in the remaining sections without loss of generality since visual atoms are extracted within each video segment independently) is further partitioned into K consecutive short-term video slices $v_1, \ldots, v_K$ (e.g., each v_k is 1-sec long). From each v_k we uniformly sample a set of frames $\tilde{I}_k^1, \ldots, \tilde{I}_k^T$ with a relatively high frequency, for example, 30 fps or 10 fps. Our task is to extract visual atoms from the video segment u.

3.1 Short-Term Point Track

Image features (corners etc.) that can be easily locked-on are automatically found [Shi and Tomasi 1994] and then tracked by using the *Kanade-Lucas-Tomasi* (*KLT*) Tracker [Birchfield 2007] for every short-term video slice v (again, we omit index k for short-term video slices in this subsection and the next one without loss of generality since short-term region tracks are extracted within each short-term video slice independently). The result is a set of N_p feature tracks, and each feature track has a trajectory $P_j^t = (x1_j^t, x2_j^t)$, where $t = 1, \ldots, T$ is the temporal index (in the unit of frames), j is the index of feature tracks, and x_1, x_2 are the image coordinates.

The KLT tracker is used because of its potent to balance reliability and speed. The KLT tracker defines a measure of dissimilarity that quantifies the change of appearance of a feature between the first and the current image frame, allowing for affine image changes. At the same time, a pure translation model of motion is used to track the selected best features through the sequence. In addition, the maximum inter-frame displacement is limited to improve the reliability and the processing speed. Alternative methods such as tracking with SIFT-based registration [Zhou et al. 2009] generally have limitations in dealing with a large amount of videos (e.g., 1358 videos with 500,000+ frames in our experiments in Section 7) due to the speed problem. In practice, we initiate the KLT tracker with 3000 initial points. In the next subsection, the extracted point tracking trajectories are used to generate short-term region tracks.

3.2 Short-Term Region Track

Each frame $\tilde{I}^t$ is segmented into a set of n_r^t homogeneous color-texture regions $r_1^t, \ldots, r_{n_r^t}^t$ by the JSeg tool developed in [Deng and Manjunath 2001]. Then from each short-term video slice v, we generate a set of N_r short-term region tracks $\mathbf{r}_1, \ldots, \mathbf{r}_{N_r}$ by the algorithm described in Figure 3. Each region track $\mathbf{r}_j$ contains a set of regions $\{r_j^t\}$, where $t = 1, \ldots, T$ is the temporal index (in the unit of frames). The basic idea is that if two regions from the adjacent frames share lots of point tracking trajectories, these two regions are considered as matched regions. To accommodate inaccurate segmentation (where a region from the frame at time t may be separated into several regions at time $t+1$, or several regions from time t may be merged at time $t+1$), we use a replication method to keep all the possible region tracks as illustrated in Figure 4. Such an approach not only retains all possible region tracks to provide rich information for constructing AVA-based codebooks in later sections, but also helps to reduce the noise from inaccurate segmentation. By treating the short-term region track as a whole unit, the influence of wrong segments from the few frames can be reduced by averaging across good segments as majorities. Finally many replications will have similar visual features and their influences in building the audio-visual codebook will be merged through the further clustering process in later parts of this section.

Note that the given STR-PTRS algorithm may miss some region tracks that enter into the screen in the middle of a short-term video slice. However such regions will still be found in the next video slice as long as they stay in the screen long enough. For those regions that enter and exit the screen

```
**Input:** A set of frames $\tilde{I}^1, \ldots, \tilde{I}^T$ from a short-term video slice $v$. Regions $r_1^t, \ldots, r_{n_r^t}^t$ for each frame $\tilde{I}^t$. A set of $N_p$ point tracks $P_j^t$, $j=1, \ldots, N_p$, $t=1, \ldots, T$.
1. Initialization: set $\mathcal{R}=\phi$, $N_r=0$.
2. Iteration: for $t=1, \ldots, T$
—Set $\mathcal{U}=\phi$.
—Get $M_{k,g}^{t|t+1}$ for each region pair $r_k^t \in \tilde{I}^t$, $r_g^{t+1} \in \tilde{I}^{t+1}$: $M_{k,g}^{t|t+1} = \sum_{j=1}^{N_p} I(P_j^t \in r_k^t) I(P_j^{t+1} \in r_g^{t+1})$.
  —If $M_{k,l^*}^{t|t+1} > H_{low}$ ($l^* = \arg\max_g M_{k,g}^{t|t+1}$), add matched region pair $(r_k^t, r_{l^*}^{t+1})$ to $\mathcal{U}$.
  —If $M_{k,l}^{t|t+1} > H_{high}$ ($l \neq l^*$), add matched region pair $(r_k^t, r_l^{t+1})$ to $\mathcal{U}$.
—Iteration: for the set of $m^t$ region pairs $(r_k^t, r_{g_1}^{t+1}), \ldots, (r_k^t, r_{g_{m^t}}^{t+1})$ in $\mathcal{U}$ starting with region $r_k^t$:
  —If there exist $m^r$ region tracks $\mathbf{r}_1, \ldots, \mathbf{r}_{m^r}$, $\mathbf{r}_j \in \mathcal{R}$ ending with $r_k^t$, replicate each $\mathbf{r}_j$ by $m^t$ times
    and append $r_{g_1}^{t+1}, \ldots, r_{g_{m^t}}^{t+1}$ to the end of each replication respectively. Set $N_r = N_r + m^t \times m^r$.
  —Else, create new tracks $\mathbf{r}_{N_r+1}, \ldots, \mathbf{r}_{Nr+m^t}$ starting with $r_k^t$ and ending with each
    $r_{g_1}^{t+1}, \ldots, r_{g_{m^t}}^{t+1}$ respectively. Set $N_r = N_r + m^t$.
3. Remove tracks in $\mathcal{R}$ with lengths shorter than $H_{long}$. Output the remaining region tracks.
```

Fig. 3. The algorithm to generate short-term region tracks. $I(\cdot)$ is the indicator function. In practice, we empirically set $H_{long} = \frac{1}{2}T$, $H_{low} = 10$, $H_{high} = \frac{1}{2}M_{k,l^*}^{t|t+1}$.

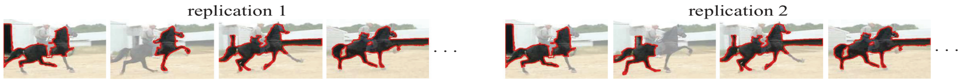

Fig. 4. An example of region track replication. In the 2^{nd} frame the horse is separated to two parts by inaccurate segmentation. We keep both two possible region tracks.

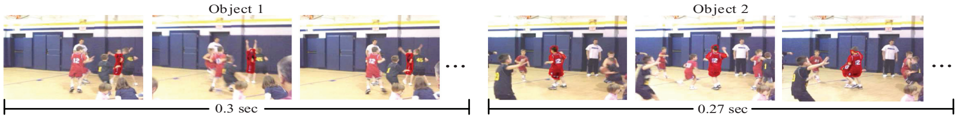

Fig. 5. Examples of fast moving objects that can only be captured with 30 fps tracking within 0.3-sec video slices. The number (e.g., 0.27 sec) below each track shows how long the track lasts.

very fast (e.g., within a video slice), they are negligible in most unconstrained videos for the purpose of semantic concept detection. Similarly, if a shot transition happens within a video slice, most region tracks during the transition may be thrown away, and the final detection performance will hardly be affected. In addition, our STR-PTRS can be extended by adding a backward checking process to overcome this problem. In other words, we start the tracking process from the last frame of the video slice and continue backward in the temporal direction.

To select the appropriate length for short-term video slices, we need to consider two aspects. The video slice needs to be short so that a good number of point tracking trajectories can be found to get region tracks. On the other hand, the longer the video slice is, the better information it retains about temporal movements in visual and audio signals. From our empirical study, a 1-sec length gives a balanced choice in general and is used in practice. However, for some videos, such as sports videos, where lots of fast-moving objects exist, a higher frame rate and a shorter tracking window are needed in order to successfully track key points of fast moving regions. Figure 5 gives examples of such fast moving objects that can only be tracked at the rate of 30 fps and the track lasts for only 0.3 sec. One future work is to design a selection method to determine the appropriate length according to different videos.

3.3 Visual Features over Region Tracks

In this section we generate visual features for the short-term region track $\mathbf{r}_j$. First, several types of visual features are extracted from each region $r^t_j \in \mathbf{r}_j$, including color moments in the HSV space (9-dim), Gabor texture (48-dim), and edge direction histogram (73-dim). These features have been shown effective in detecting generic concepts [Yanagawa et al. 2006]. We concatenate these features into a 130-dim vector $\tilde{\mathbf{f}}_{j,vis}{}^t$ and then average $\tilde{\mathbf{f}}^t_{j,vis}$ across time $t = 1, \ldots, T$ to obtain a 130-dim feature vector $\mathbf{f}_{j,vis}$ for the region track $\mathbf{r}_j$. $\mathbf{f}_{j,vis}$ describes the overall visual characteristics of $\mathbf{r}_j$. In addition, optical flow vectors are calculated over every pixel of each frame $\tilde{I}^t_j$ using the Lucas-Kanade algorithm [Lucas and Kanade 1981], where for each pixel $(x1^t_j, x2^t_j)$ a motion vector $\mathbf{m}(x1^t_j, x2^t_j)$ is obtained. Then for each region $r^t_j \in \mathbf{r}_j$, a 4-dim feature $\tilde{\mathbf{f}}_{j,mt}{}^t$ is computed, where each bin corresponds to a quadrant in the 2D motion space, and the value for this bin is the average speed of motion vectors moving along directions in this quadrant. Then we average $\tilde{\mathbf{f}}_{j,mt}{}^t$ across $t = 1, \ldots, T$ to obtain a motion feature $\mathbf{f}_{j,mt}$ for $\mathbf{r}_j$. $\mathbf{f}_{j,mt}$ describes the overall moving speed and direction of $\mathbf{r}_j$. The coarse 4-bin granularity is empirically chosen since for concept detection fine granularity of motion directions can be very noisy; for example, an animal can move towards any direction. The coarse description of motion speed and direction gives relatively robust performance in general. In addition, we generate a spatial feature vector $\mathbf{f}_{j,loc}$ describing the spatial location information for each $\mathbf{r}_j$. $\mathbf{f}_{j,loc}$ has 3 dimensions, corresponding to the horizontal and vertical coordinates of the center of $\mathbf{r}_j$ and the size of $\mathbf{r}_j$, all being averaged results across the tracked regions in $\mathbf{r}_j$.

Note that more visual features can be extracted to describe short-term region tracks, such as local descriptors like SIFT [Lowe 2004]. In our experiments, we construct the Bag-of-Features (BoF) histogram $\mathbf{f}_{j,sift}$ for the region track $\mathbf{r}_j$ based on a codebook generated by clustering SIFT features from a set of training videos, following the recipe of [Grauman and Darrel 2005]. These SIFT descriptors are calculated over interest points detected with the Hessian-Affine algorithm. However, for concept detection over our consumer videos, $\mathbf{f}_{j,sift}$ can not compete with the global regional $\mathbf{f}_{j,vis}$ in general, as will be demonstrated in both the experiments of Section 7 and Jiang [2010]. This phenomenon intuitively confirms how challenging this consumer collection is. Due to the large diversity in the visual content, there is very little repetition of objects or scenes in different videos, even those from the same concept class. In such a case, it is hard to exert the advantage of local descriptors like SIFT for local point matching/registration. Nonetheless, $\mathbf{f}_{j,sift}$ can still be used as additional descriptors to complement the global regional visual features.

3.4 Long-Term Linking to Generate Visual Atoms

By now, for an input video segment u, over each short-term video slice v_k, $k = 1, \ldots, K$, we have a set of N^k_r short-term region tracks $\mathbf{r}^k_1, \ldots, \mathbf{r}^k_{N^k_r}$. In this section, we connect these short-term region tracks from adjacent video slices together into long-term visual atoms. Such linking is based on the low-level visual feature $\mathbf{f}^k_{j,vis}$ and the location information $\mathbf{f}^k_{j,loc}$ of each region track $\mathbf{r}^k_j$. Figure 6 gives the algorithm used for long-term linking. Figure 7 shows some example visual atoms after long-term linking. From the figure, over some consistent large regions we can get good visual atoms extracted to capture salient regions in the video. Actually, over the example "wedding" video, there are over 100 visual atoms extracted in total, and in the figure we only show a few of them. About 80% of the remaining visual atoms are relatively short (e.g., lasting for two to three seconds). This is due to the dramatic content change and motion in such unconstrained consumer videos.

```
Input: Short-term video slices $v_1, \dots, v_K$ from a video segment $u$. $N_r^k$ short-term region tracks $\mathbf{r}_1^k, \dots, \mathbf{r}_{N_r^k}^k$ for each short-term slice $v_k$. $\mathbf{f}_{j,vis}^k$ and $\mathbf{f}_{j,loc}^k$ associated with each $\mathbf{r}_j^k$.
1. Initialization: set $\mathcal{R} = \phi$, $N_{vis} = 0$.
2. Iteration: for $k = 1, \dots, K$
—Set $\mathcal{U} = \phi$.
—Calculate the pairwise distance $D_{vis}(\mathbf{r}_i^k, \mathbf{r}_j^{k+1})$ and $D_{loc}(\mathbf{r}_i^k, \mathbf{r}_j^{k+1})$ between region tracks $\mathbf{r}_i^k \in v_k$ ($i = 1, \dots, N_r^k$) and region tracks $\mathbf{r}_j^{k+1} \in v_{k+1}$ ($j = 1, \dots, N_r^{k+1}$) over $\mathbf{f}_{vis}$ and $\mathbf{f}_{loc}$, respectively.
—For each region track $\mathbf{r}_i^k \in v_k$:
  —If $D_{vis}(\mathbf{r}_i^k, \mathbf{r}_{l^*}^{k+1}) < H_{vis}^{low}$ & $D_{loc}(\mathbf{r}_i^k, \mathbf{r}_{l^*}^{k+1}) < H_{loc}$ (where $l^* = \arg\min_j D_{vis}(\mathbf{r}_i^k, \mathbf{r}_j^{k+1})$), add matched pair of region tracks $(\mathbf{r}_i^k, \mathbf{r}_{l^*}^{k+1})$ to $\mathcal{U}$.
  —If $D_{vis}(\mathbf{r}_i^k, \mathbf{r}_l^{k+1}) < H_{vis}^{high}$ & $D_{loc}(\mathbf{r}_i^k, \mathbf{r}_l^{k+1}) < H_{loc}$ ($l \neq l^*$), add matched pair of region tracks $(\mathbf{r}_i^k, \mathbf{r}_l^{k+1})$ to $\mathcal{U}$.
—Iteration: for $m^k$ pairs of tracks $(\mathbf{r}_j^k, \mathbf{r}_{g_1}^{k+1}), \dots, (\mathbf{r}_j^k, \mathbf{r}_{g_{m^k}}^{k+1})$ in $\mathcal{U}$ starting with track $\mathbf{r}_j^k$:
  —If there exist $m^a$ visual atoms $\mathcal{A}_1^{vis}, \dots, \mathcal{A}_{m^a}^{vis}$, $\mathcal{A}_i^{vis} \in \mathcal{R}$ that end with region track $\mathbf{r}_j^k$, replicate each $\mathcal{A}_i^{vis}$ by $m^k$ times, and extend each visual atom replication by appending region tracks $\mathbf{r}_{g_1}^{k+1}, \dots, \mathbf{r}_{g_{m^k}}^{k+1}$ to the end of each replication respectively. Set $N_{vis} = N_{vis} + m^k \times m^a$.
  —Else, create new visual atoms $\mathcal{A}_{N_{vis}+1}^{vis}, \dots, \mathcal{A}_{N_{vis}+m^k}^{vis}$ starting with region track $\mathbf{r}_j^k$ and ending with each $\mathbf{r}_{g_1}^{k+1}, \dots, \mathbf{r}_{g_{m^k}}^{k+1}$ respectively. Set $N_{vis} = N_{vis} + m^k$.
3. Output the visual atoms in $\mathcal{R}$.
```

Fig. 6. The algorithm for long-term linking to generate visual atoms. We empirically set $H_{vis}^{high} = 0.25$, $H_{vis}^{low} = 0.2$, $H_{loc} = 0.15|Ig|$ where $|Ig|$ is the size of a video frame.

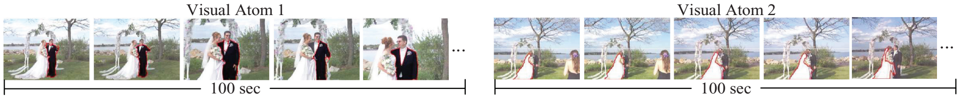

Fig. 7. Examples of the final visual atoms extracted from a "wedding" video.

3.5 Refinement through Clustering

For each input video segment u, we have a set of N_{vis} visual atoms extracted as $\mathcal{A}_1^{vis}, \dots, \mathcal{A}_{N_{vis}}^{vis}$. Each visual atom corresponds to a set of visual regions that are spatially continuous with consistent visual appearances through time. Visual atoms can capture individual objects, parts of objects, or combinations of multiple objects, through time. However, such a representation is not temporally complete, that is, a visual atom may capture a consistent region over discontinued time windows. The incomplete tracking results can be attributed to the challenging conditions for visual tracking in unconstrained videos. In this section, we refine raw visual atoms through clustering, where we group visually similar visual atoms together as one entity for later usage. The goal is that such clustered groups can merge visual atoms that describe the same visual regions (but are tracked separately at different times) together. In practice, the hierarchical clustering algorithm is conducted based on visual features $\mathbf{f}_{vis}$. Figure 8 shows examples of clustering results. Visual atoms in the same cluster can be regions of the same object from different times of different areas but broken to separate tracks due to tracking gaps in time or over segmentation in space. This process has the effect of refining visual atoms and removing redundant visual atoms.

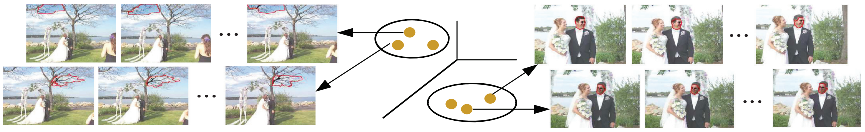

Fig. 8. Examples of clustered visual atoms.

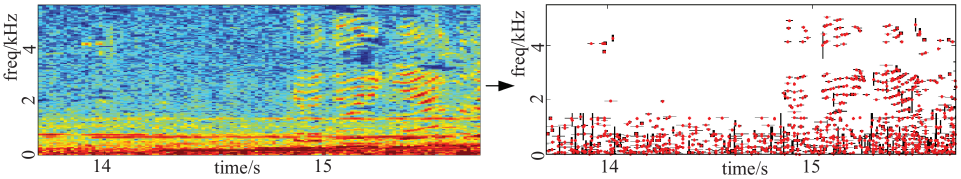

Fig. 9. An example of the original audio signal and the corresponding time-frequency bases.

4. AUDIO ATOM EXTRACTION

In this section, we describe the process to extract audio atoms from the sound track corresponding to each video segment u where visual atoms are generated.

4.1 Extracting Time-Frequency Bases

We represent the audio sound using a matching pursuit decomposition [Mallat and Zhang 1993]. The bases used for MP are Gabor functions, which are Gaussian-windowed sinusoids. The Gabor function is evaluated at a range of frequencies covering the available spectrum, scaled in length (trading time resolution for frequency resolution), and translated in time. The created functions form a dictionary, which possesses a continuum of time-frequency localization properties. The length scaling creates long functions with narrowband frequency resolution, and short functions (well-localized in time) with wideband frequency resolution. This amounts to a modular Short-Time Fourier Transform representation, with analysis windows of variable length. During MP analysis, functions are selected in a greedy fashion to maximize the energy removed from the signal at each iteration, resulting in a sparse representation. This process has the effect of denoising the signal while retaining information about the most important parts of the signal. The Matching Pursuit Toolkit [Krstulovic and Gribonval 2006], an efficient implementation of the algorithm, is used. The dictionary contains functions at eight length scales, incremented by powers of two. For data sampled at 16 kHz, this corresponds to durations ranging from 2 to 256 ms. These are each translated in increments of one eighth of the function length, over the duration of the signal.

To ensure coverage of the audio activity in each short-term window, we extract a fixed number of functions (500) from each window. We then prune this set of functions with postprocessing based on psychoacoustic masking principles [Petitcolas 2003]. This emulates the perceptual effect by which lower energy functions close in frequency to higher-energy signal cannot be detected by human hearing. We retain the 70% of the functions with the highest perceptual prominence relative to their local time-frequency neighborhood. This emphasizes the most salient functions, and removes less noticeable ones. Figure 9 shows the original audio signal and the extracted time-frequency bases.

4.2 Locate Audio Onsets

Using a detection function built from the positions, lengths, and amplitudes of the extracted time-frequency bases, we determine a set of times at which onsets of energy probably occurred in the audio.

To do this we sum the envelopes of all the functions that are retained after the pruning process, and this is our onset detection function. We then use two different approaches to select peaks from this function. First we select all peaks from the detection function and keep those that are above a threshold (some percentage above the local mean). This detects most sharp onsets. To detect softer but still significant onsets, we smooth the original detection function by low-pass filtering, and then repeat the peak-picking process. The two sets of potential onset times are then combined. Additionally, we prune the final set of onset times by requiring that onsets be no closer together than 50 ms. We therefore remove small peaks that are closer to a larger peak than 50 ms.

4.3 Audio Atoms and Features

For each energy onset, we collect all those time-frequency bases whose center times fall within a short window around the onset time, and we throw away all other time-frequency bases from the audio signal. Then we reconstruct the audio signal around each onset separately, using only those remaining few time-frequency bases. After that, we generate a coarsely-binned mel-frequency spectrogram representation of the short window around each onset. The dimensions of the spectrogram patches are 20 frequency bands by 30 frames in time (the frames are standard MFCC frames: 25 ms windows with 10 ms hops). Compared to conventional features like MFCCs, these new features are designed to be relatively invariant to background noise and to variations in acoustic channel characteristic, due to the focus on energy onsets. Such audio features around energy onsets provide a natural domain for segmenting the representation into portions associated with distinct sound sources. This ability gives the opportunity to study moderate-level audio-visual correlation.

We perform PCA on the audio spectrograms collected from all onsets from all training videos and keep the first 20 principal components. Then for each video segment, we cluster the extracted audio atoms in the 20-dim PCA space. The centers of the few largest clusters are used as the final audio atoms to correlate with the visual atoms. Typically each video segment contains about 10 audio atoms.

5. JOINT AUDIO-VISUAL CODEBOOK CONSTRUCTION

For each video segment u, we have N_{vis} visual atoms $\mathcal{A}_1^{vis}, \ldots, \mathcal{A}_{N_{vis}}^{vis}$ and N_{aud} audio atoms $\mathcal{A}_1^{aud}, \ldots, \mathcal{A}_{N_{aud}}^{aud}$. We associate each audio atom with each visual atom to obtain a cross-product number of AVAs: $\mathcal{A}_i^{vis\text{-}aud}$, $i = 1, \ldots, N_{vis} \times N_{aud}$. Each $\mathcal{A}_i^{vis\text{-}aud}$ contains a long-term region track associated with a global visual feature $\mathbf{f}_{i,vis}$ (d_{vis}-dim), a local visual feature $\mathbf{f}_{i,sift}$ (d_{sift}-dim), a spatial feature $\mathbf{f}_{i,loc}$ (d_{loc}-dim), a motion feature $\mathbf{f}_{i,mt}$ (d_{mt}-dim), and an audio feature $\mathbf{f}_{i,audio}$ (d_{audio}-dim). We can concatenate different types of features into various multi-modal vectors, based on which various multi-modal codebooks can be constructed.

A video concept detection task usually has the following formulation: keyframes are sampled from each video segment and are annotated with binary labels. In our experiment, one keyframe I_l is sampled from each video segment u_l. A binary label $y_{I_l}^m = 1$ or -1 is assigned to each keyframe I_l to indicate the occurrence or absence of a concept C^m ($m = 1, \ldots, M$) in the video segment u_l. Based on this structure, we extract a set of AVAs over each video segment, and then we use these AVAs to construct a discriminative joint audio-visual codebook for each concept C^m.

Each video segment u_l can be treated as a "bag-of-AVAs"; that is, it consists of a set of AVAs generated from the previous sections, and each AVA is an instance in the video-segment bag. Thus y_I is the label over the bag rather than over instances. For a semantic concept C^m, it is sensible to assume that a "positive" bag u_l (with $y_{I_l}^m = 1$) must have at least one of its instances being "positive"; for example, a positive video segment for concept "animal" must have at least one "animal" AVA. On the other hand, a "negative" bag u_l (with $y_{I_l}^m = -1$) does not have any "positive" instance. This problem is known as MIL [Chen and Wang 2004; Maron and Lozano-Pérez 1998; Yang et al. 2006] in the literature.

With different concatenations of $\mathbf{f}_{i,vis}$, $\mathbf{f}_{i,loc}$, $\mathbf{f}_{i,sift}$, $\mathbf{f}_{i,mt}$, and $\mathbf{f}_{i,audio}$, various multi-modal features can be generated to describe an AVA $\mathcal{A}_i^{vis\text{-}aud}$. Assume that we have a combined d-dim feature space. For each concept C^m, we repeat an MIL-type procedure P_m-times in order to obtain P_m discriminative prototypes $(\mathbf{f}_p^{m*}, \mathbf{w}_p^{m*})$, $p=1, \ldots, P_m$, consisting of a prototype point (or centroid) $\mathbf{f}_p^{m*}=[f_{p1}^{m\,*}, \ldots, f_{pd}^{m\,*}]^T$ in the d-dim feature space, and the corresponding weights for each dimension $\mathbf{w}_p^{m*}=[w_{p1}^{m\,*}, \ldots, w_{pd}^{m\,*}]^T$.

Among the flavors of MIL objective functions, the Diverse Density (DD) is one that fits our objective of finding discriminative prototypes and also with efficient inference algorithm available [Chen and Wang 2004] via EM. In the rest of Section 5, we omit subscripts m, p w.l.o.g., as each $\mathbf{f}^*$ is independently optimized for different concepts over different video segment bags $l \in \{1, \ldots, L\}$ and different instances $j \in \{1, \ldots, N_l\}$ in each bag u_l. The DD objective function for a bag u_l is:

$$Q_l = (1 + y_{I_l})/2 - y_{I_l} \prod\nolimits_{j=1}^{N_l} \left(1 - e^{-||\mathbf{f}_{lj} - \mathbf{f}^*||^2_{\mathbf{w}^*}}\right), \tag{1}$$

where $\mathbf{f}_{lj}$ is the feature vector of the j-th AVA instance, and $||\mathbf{f}||_{\mathbf{w}}$ is the weighted 2-norm of vector $\mathbf{f}$ by $\mathbf{w}$, *i.e.*, $||\mathbf{f}||_{\mathbf{w}} = (\sum_{i=1}^{d} (f_i w_i)^2)^{\frac{1}{2}}$. For a positive bag u_l, Q_l will be close to 1 when $\mathbf{f}^*$ is close to any of its instances, and Q_l will be small when $\mathbf{f}^*$ is far from all its instances. For a negative bag u_l, Q_l will be large when $\mathbf{f}^*$ is far from all its instances. By aggregating Equation (1) over all bags the optimal $\mathbf{f}^*$ will be close to instances in positive bags and far from all instances in negative bags.

For each positive video segment bag u_l, there should be at least one AVA to be treated as a positive sample to carry the label of that bag. This instance, denoted by $J(u_l)$, is identified as the closest instance to the prototype $\mathbf{f}^*$ and is given by Equation (2). For each negative bag u_l (with $y_{I_l} = -1$), on the other hand, all instances are treated as negative samples, whose contributions to Q_l are all preserved.

$$J(u_l) = \arg\max\nolimits_{j=1}^{N_l} \left\{ \exp\left[-||\mathbf{f}_{lj} - \mathbf{f}^*||^2_{\mathbf{w}^*} \right] \right\}. \tag{2}$$

This leads to the max-ed version of Equation (1) on positive bags:

$$Q_l = \begin{cases} e^{-||\mathbf{f}_{lJ(u_l)} - \mathbf{f}^*||^2_{\mathbf{w}^*}} & , \ y_{I_l} = 1 \\ \prod_{j=1}^{N_l} (1 - e^{-||\mathbf{f}_{lj} - \mathbf{f}^*||^2_{\mathbf{w}^*}}) & , \ y_{I_l} = -1. \end{cases} \tag{3}$$

The DD function in Equation (3) is used to construct an objective function Q over all bags, $Q = \prod_{l=1}^{L} Q_l$. Q is maximized by an EM algorithm [Chen and Wang 2004]. We use each instance in each positive bag to repeatedly initiate the DD-optimization process presented above, and prototypes with DD values smaller than a threshold H_{dd} (that equals to the mean of DD values of all learned prototypes) are excluded. Such a prototype learning process is conducted for each semantic concept independently, and the final learned prototypes construct a codebook to describe the discriminative multimodal characteristics of each individual concept.

6. CLASSIFICATION WITH JOINT A-V CODEBOOKS

For each concept C^m, the learned prototypes form a codebook to describe its discriminative characteristics, each prototype corresponding to a codeword. These codewords span a codebook-based feature space to represent AVAs. For an AVA with a long-term region track $\mathbf{r}_j$ and a feature $\mathbf{f}_j$, it can be mapped to each codeword $(\mathbf{f}_p^{m*}, \mathbf{w}_p^{m*})$ by the weighted norm-2 distance $||\mathbf{f}_j - \mathbf{f}_p^{m*}||^2_{\mathbf{w}_p^{m*}}$. Accordingly, each video segment u can be mapped to each codeword by using the minimum distance $D(u, \mathbf{f}_p^{m*})_{\mathbf{w}_p^{m*}} = \min_{\mathbf{r}_j \in u} \{||\mathbf{f}_j - \mathbf{f}_p^{m*}||^2_{\mathbf{w}_p^{m*}}\}$. Then the video segment u can be represented by a codebook-based feature

```
Input: Training set $\mathcal{D} = \{(u_1, y_{I_1}), \ldots, (u_L, y_{I_L})\}$. Each $u_l$ is represented by several codebook-based features learned with various combinations of $\mathbf{f}_{vis}$, $\mathbf{f}_{mt}$, $\mathbf{f}_{sift}$, $\mathbf{f}_{loc}$, and $\mathbf{f}_{audio}$.
1. Initialization: set sample weights $\sigma_l^1 = 1/2L^+$ or $1/2L^-$ for $y_{I_l} = 1$ or $-1$, respectively, where $L^+$ ($L^-$) is the number of positive (negative) samples; set final decisions $H^1(u_l) = 0$, $l = 1, \ldots, L$.
2. Iteration: for $\tau = 1, \ldots, \Gamma$
—Get training set $\tilde{\mathcal{D}}^\tau$ by sampling $\mathcal{D}$ according to weights $\sigma_l^\tau$; Train an SVM over $\tilde{\mathcal{D}}^\tau$ using the
   $k$-th type of feature. Get the corresponding $q_k^\tau(u_l) = p_k^\tau(y_{I_l} = 1|u_l)$, $l = 1, \ldots, L$.
—Set $h_k^\tau(u_l) = \frac{1}{2} \log \left[ q_k^\tau(u_l) / \left(1 - q_k^\tau(u_l)\right) \right]$; Choose the optimal $h^{\tau,*}(\cdot) = h_k^\tau(\cdot)$ with the minimum
   error $\epsilon^{\tau,*} = \epsilon_k^\tau$, $\epsilon_k^\tau = \sum_{l=1}^N \sigma_l^\tau e^{-y_{I_l}(H^\tau(u_l) + h_k^\tau(u_l))}$, $\epsilon_k^\tau < \epsilon_j^\tau$ if $j \neq k$.
—Update weights: $\sigma_l^{\tau+1} = \sigma_l^\tau e^{-y_{I_l} h^{\tau,*}(u_l)}$, $l = 1, \ldots, L$, and re-normalize so that $\sum_{l=1}^L \sigma_l^{\tau+1} = 1$.
—Update $H^{\tau+1}(u_l) = H^\tau(u_l) + h^{\tau,*}(u_l)$ for $l = 1, \ldots, L$.
```

Fig. 10. The algorithm to construct concept detectors by selectively using different codebooks. 10 iterations are empirically taken in our experiments ($\Gamma = 10$).

$\mathbf{D}^m(u) = [D(u, \mathbf{f}_1^{m*})_{\mathbf{w}_1^{m*}}, \ldots, D(u, \mathbf{f}_{P_m}^{m\,*})_{\mathbf{w}_{P_m}^{m\,*}}]$, base on which classifiers like SVMs can be trained for concept detection.

By using different combinations of $\mathbf{f}_{vis}$, $\mathbf{f}_{sift}$, $\mathbf{f}_{loc}$, $\mathbf{f}_{mt}$, and $\mathbf{f}_{audio}$, various codebooks can be generated in different multi-modal feature spaces. In general, different types of codebooks have uneven advantages at detecting different concepts. We selectively choose the optimal types of codebooks to use by adopting a boosting feature selection framework similar to Tieu and Viola [2000]. The Real AdaBoost method [Friedman et al. 2000] is used where during each iteration, an optimal codebook is selected to construct an SVM classifier as the weak learner, and the final detector is generated by adding up weak learners from multiple iterations. The boosting algorithm is summarized in Figure 10.

7. EXPERIMENTS

We evaluate our algorithm over Kodak's consumer benchmark videos [Loui et al. 2007], which contains 1358 videos from real consumers. 5166 keyframes are uniformly sampled from the videos for every 10 seconds, and are labeled to 21 concepts that are of great interest based on a real user study. The concepts fall into several broad categories including activities (e.g., sports), occasions (e.g., wedding), locations (e.g., playground), scenes (e.g., sunset), or objects in the scene (e.g., boat).

We separate the entire data set into two subsets: 60% videos, that is, 813 videos, are randomly sampled as the training data; and the rest 40% videos are used for testing. This is a multilabel dataset, that is, each keyframe can have multiple concept labels. One-vs.-all classifiers are trained for detecting each individual concept, and the *average precision* (*AP*) and *mean average precision* (*MAP*) [Smeaton et al. 2006] are used as performance measures. To extensively evaluate the proposed audio-visual analysis framework, we experiment on two different concept detectors using the audio-visual atomic representation: (1) Visual Atoms with MIL codebook construction (VA-MIL), where the visual-codebook-based features are directly used to train SVM detectors; and (2) AVA with MIL codebook construction and Boosting feature selection (AVA-MIL-Boosting), where different types of codebooks are generated and selectively used via Boosting. In addition, we compare VA-MIL with two state-of-the-art static-region-based image categorization approaches that also use MIL, that is, DD-SVM [Chen and Wang 2004] and ASVM-MIL [Yang et al. 2006]. For static-region-based methods, each video-segment bag u contains a set of static regions that come from the center frame of each short-term video slice v in this bag. DD-SVM learns visual codebooks with static bags using MIL for individual concepts, and codebook-based features are generated to train SVMs. ASVM-MIL directly builds an asymmetrical SVM over the static regions under the MIL setting. No temporal tracking is involved in both of these two approaches.

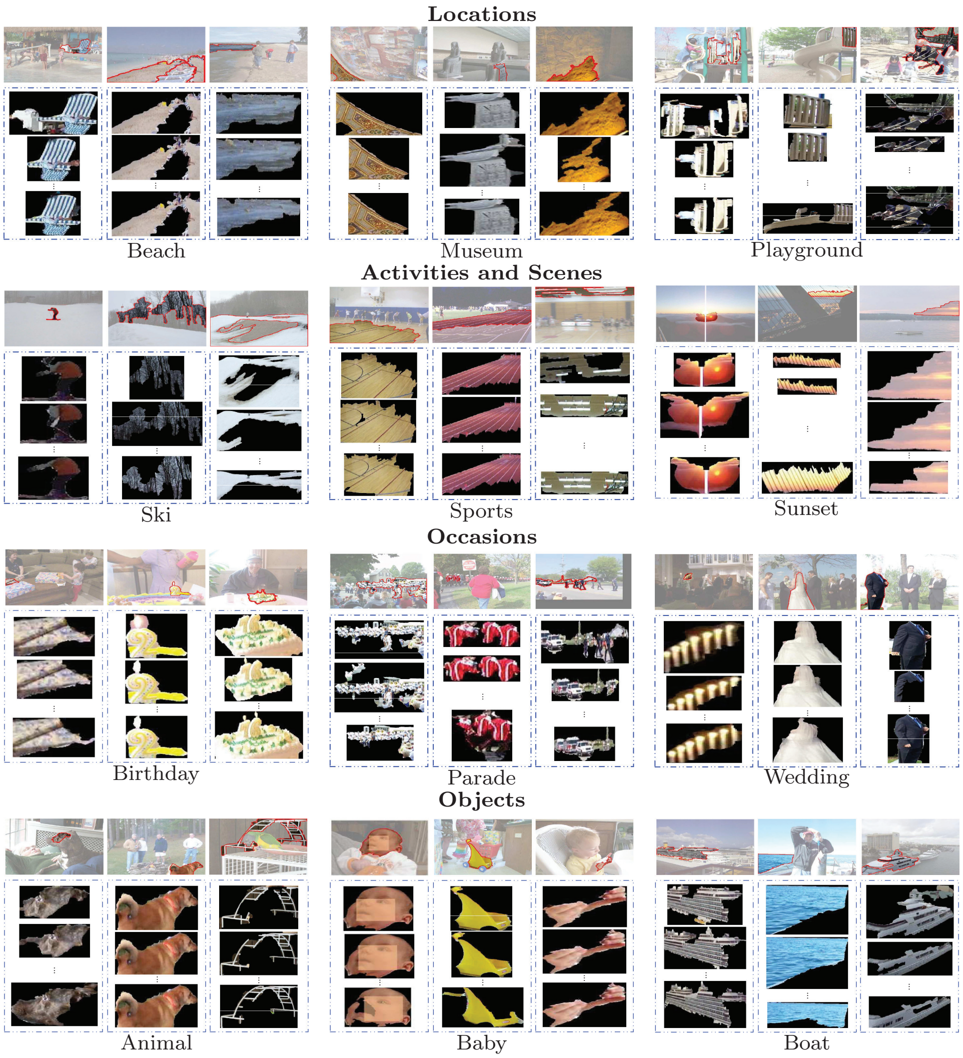

Fig. 11. Example prototypes learned with audio-visual atoms. Doted blue boxes show the corresponding region track prototypes, where images on the top show example frames where region tracks are extracted.

7.1 Codebook Visualization

We first show examples of the discriminative prototypes learned in various types of codebooks. Such visualization helps us to subjectively and intuitively evaluate different approaches. To visualize each prototype ($\mathbf{f}^*$, $\mathbf{w}^*$), we calculate the distances between all training AVA instances and this prototype. Then the AVA with the minimum distance is considered as the most appropriate example to visualize this prototype. In addition, prototypes learned for each concept can be ranked according to the DD values Q in descending order. The higher rank a prototype has, the better the prototype describes the discriminative characteristics of this concept. Figure 11 gives some example prototypes (ranked within top 50) extracted based on audio-visual atoms. From Figure 11, the audio-visual-atom-based

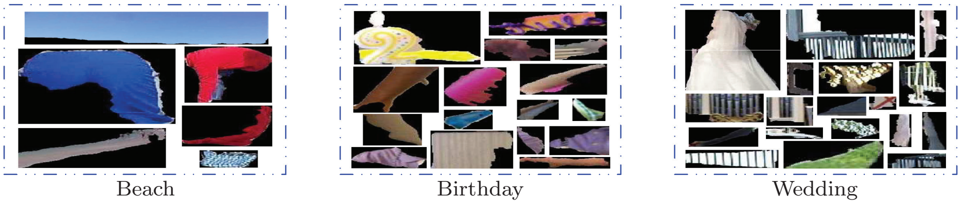

Fig. 12. Examples of static prototypes by DD-SVM. The static codebook contains lots of noise, for example, fragments of human clothes, which causes severe degradation of classification performance.

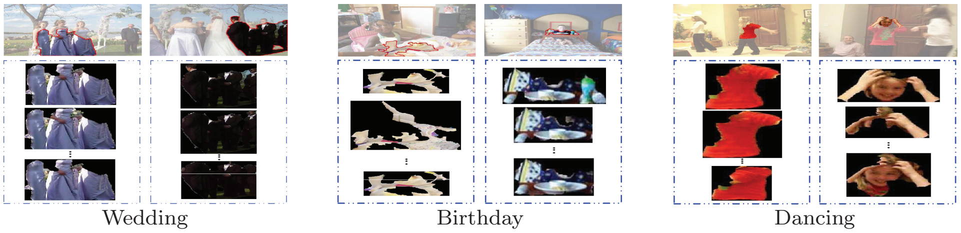

Fig. 13. Example prototypes learned with audio-visual atoms for concepts that are expected to have strong cues in both audio and visual aspects. These prototypes are not discovered by visual-only codebooks.

prototypes are very reasonable. For example, in Location category we get water, sand, and beach facility as representative patterns to describe "beach" concept; in Activity-and-Scene category, we get the white snow, bald white trees, and the athlete as representative patterns to describe "ski" concept; in Occasion category, we get wedding gown, black suit, and wedding candles as representative patterns to describe "wedding" concept; and in Object category we get the baby face, baby hand, and baby toys as representative patterns to describe "baby" concept.

In comparison, Figure 12 gives some example prototypes learned by the static-region-based DD-SVM algorithm. In later experiments we will see that our method significantly outperforms static-region-based approaches. The prototype visualization helps to explain such results. The static DD-SVM gets very noisy prototypes in general, for example, many fragments of human clothes are extracted as representative patterns. Although some good prototypes can also be obtained, the performance suffers from the noisy ones a lot. The results also confirm our motivation that region tracks are more noise-resistent for video concept detection than static regions.

By adding audio features to visual atoms, salient audio-visual patterns can be discovered by the audio-visual codebook for concepts that are expected to have strong cues in both audio and visual aspects. Figure 13 gives some example prototypes learned by using AVAs with the concatenation of $\mathbf{f}_{vis}$ and $\mathbf{f}_{audio}$. These prototypes are salient for concept detection, but are not captured by visual-atom-based codebooks. For example, those salient patterns about the tableware with a piece of birthday cake inside can be discovered by considering audio and visual features jointly but can not be extracted by using visual features alone. This is because tableware also appears in many other videos visually, and only when combined with the background birthday music can the tableware generate salient audio-visual cues to describe "birthday" videos. Similarly, body parts of a dancing person can be discovered by using audio-visual atoms but are missed by using visual features alone, since only when combined with background music can the body parts form salient audio-visual cues to describe "dancing" videos.

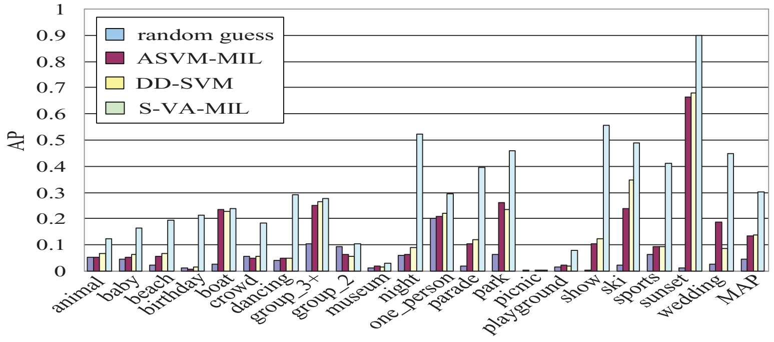

Fig. 14. Comparison of visual atoms with static-region-based methods.

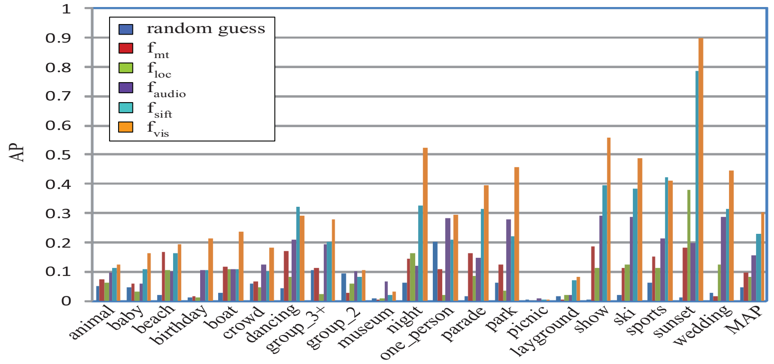

Fig. 15. Comparison of AVA-MIL with individual $\mathbf{f}_{vis}$, $\mathbf{f}_{loc}$, $\mathbf{f}_{sift}$, $\mathbf{f}_{mt}$, and $\mathbf{f}_{audio}$.

7.2 Performance of Concept Detection

We compare AP and MAP of different algorithms for concept detection. All methods use the RBF kernel and the multiple-parameter technique [Chang et al. 2007] instead of tuning parameters by cross-validation. This is based on findings in Chang et al. [2007], that is, over the challenging consumer videos, parameter tuning tends to over fit due to the large diversity of the video content.

7.2.1 *Region Tracks vs. Static Regions.* As described in Section 6, each visual atom is associated with a visual feature vector $\mathbf{f}_{vis}$. For a static region, we can also extract a visual feature $\mathbf{f}_{vis}$. Figure 14, shows the per-concept AP and MAP comparison of VA-MIL, DD-SVM, and ASVM-MIL, by using $\mathbf{f}_{vis}$. The results from random guess are also shown for comparison. From the figure, our VA-MIL consistently outperforms other methods over every concept, and significant performance improvements, that is, over 120% MAP gain on a relative basis, can be achieved compared to both DD-SVM and ASVM-MIL. This phenomenon confirms that static regions segmented from unconstrained videos are very noisy. Our visual atoms can significantly reduce the noise by not only extracting robust and trackable regions, but also averaging out the outlier noise through the entire tracked sequence.

7.2.2 *AVA-MIL with Multimodal Features.* Figure 15 gives the performance comparison of our AVA-MIL by using different codebooks generated from individual $\mathbf{f}_{vis}$, $\mathbf{f}_{mt}$, $\mathbf{f}_{sift}$, $\mathbf{f}_{loc}$, and $\mathbf{f}_{audio}$. From the result, $\mathbf{f}_{mt}$ performs badly because of the low-quality motion in unconstrained videos and the lack

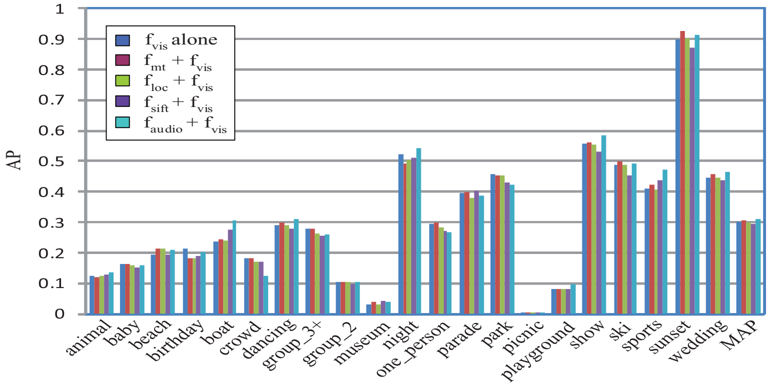

Fig. 16. Comparison of AVA-MIL with different combinations of $\mathbf{f}_{vis}$, $\mathbf{f}_{loc}$, $\mathbf{f}_{sift}$, $\mathbf{f}_{mt}$, and $\mathbf{f}_{audio}$.

of discriminative power of motion alone for concept detection. For example, the moving speed and direction of a person can not discriminate "one person" videos. Also, $\mathbf{f}_{loc}$ does not perform well due to the lack of discriminative power of spatial location alone for concept detection. In general, audio feature $\mathbf{f}_{audio}$ alone can not compete with visual feature $\mathbf{f}_{vis}$ or $\mathbf{f}_{sift}$, since most of the 21 concepts are visual-oriented. However, $\mathbf{f}_{audio}$ works very well over "museum." The visual content of "museum" videos is very diverse while the audio sound is relatively consistent, for example, the sound of people talking and walking in a large quiet indoor room. The global $\mathbf{f}_{vis}$ outperforms local $\mathbf{f}_{sift}$ over most of the concepts except for "dancing" and "sports." This is because of the challenging condition to detect unconstrained concepts in this Kodak's video collection. Due to the large diversity in the visual content, there is very little repetition of objects or scenes in different videos, even those from the same concept class. In such a case, it is hard to exert the advantage of local descriptors like SIFT for local point matching/registration.

Figure 16 gives the AP and MAP comparison of AVA-MIL using different combinations of $\mathbf{f}_{vis}$, $\mathbf{f}_{loc}$, $\mathbf{f}_{sift}$, $\mathbf{f}_{mt}$, and $\mathbf{f}_{audio}$. Compared with individual $\mathbf{f}_{vis}$, both $\mathbf{f}_{vis} + \mathbf{f}_{mt}$ and $\mathbf{f}_{vis} + \mathbf{f}_{audio}$ have slight improvements in terms of MAP. By adding the noisy motion features ($\mathbf{f}_{vis} + \mathbf{f}_{mt}$), most concepts get worse or unchanged performances except for "beach" and "sports." This is reasonable since "sports" videos often have fast moving athletes, and "beach" videos have large stable regions like water and sand that do not move. $\mathbf{f}_{vis} + \mathbf{f}_{loc}$ gets performance degradation over most concepts due to the lack of discriminative power of $\mathbf{f}_{loc}$ in detecting the set of concepts we experiment on. $\mathbf{f}_{vis} + \mathbf{f}_{sift}$ does not give overall improvement either. $\mathbf{f}_{vis} + \mathbf{f}_{audio}$ has clear improvements over 12 concepts, for example, "boat" and "sports" get 28.7% and 15.9% AP gains, respectively. However, we also have noticeable AP degradation over some concepts like "crowd" and "park," because the regional color and texture features are much more powerful in detecting these concepts than audio features. The results also indicate the uneven strengths of different modalities in detecting different concepts. Therefore, as will be shown in Section 7.2.3, a more rigorous approach in selecting optimal features from different modalities is needed.

7.2.3 *AVA-MIL-Boosting with Multimodal Features.* Figure 17 gives the performance of multimodal AVA-MIL-Boosting. Also, we compare with a straightforward fusion approach, where SVMs trained using codebooks generated from $\mathbf{f}_{vis}$, $\mathbf{f}_{vis} + \mathbf{f}_{mt}$, and $\mathbf{f}_{vis} + \mathbf{f}_{audio}$, respectively, are averaged to give the final detection results. From the figure, we can see that by selectively using the optimal types of codebooks for detecting different individual concepts, the multimodal AVA-MIL-Boosting can

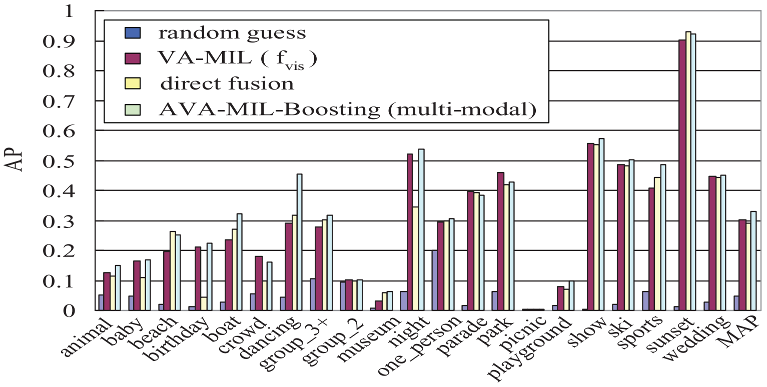

Fig. 17. Comparison of multimodal AVA-MIL-Boosting, VA-MIL with $\mathbf{f}_{vis}$, and direct fusion.

improve the detection performance over most of the concepts (17 out of 21) compared to VA-MIL. Significant AP gains are achieved (on a relative basis) for "animal" by 20.9%, "beach" by 28.3%, "boat" by 35.4%, "crowd" by 11.7%, "group of three or more" by 14.6%, "dancing" by 56.7%, "museum" by 106.3%, "playground" by 20.7%, and "sports" by 18.9%. The overall MAP is improved by 8.5%. In comparison, without feature selection, the direct fusion method can not improve the overall MAP by simply adding up classifiers from different modalities, which is consistent with results in Chang et al. [2007].

8. CONCLUSION

We study audio-visual analysis in unconstrained videos by extracting atomic representations. Visual atoms are extracted by a hierarchical framework where in the first step an STR-PTRS algorithm is developed to track consistent visual regions within short-term video slices and then in the second step the short-term region tracks are connected into visual atoms by visual linking. At the same time, the corresponding audio soundtrack is decomposed to time-frequency bases, according to which audio energy onsets are located. Audio atoms are reconstructed from the time-frequency bases around audio onsets. Visual atoms are associated with audio atoms to generate AVAs. Regional visual features and spectrogram audio features are assigned to AVAs. Joint audio-visual codebooks are constructed on top of AVAs to capture salient audio-visual patterns for effective concept detection. Our method provides a balanced choice for audio-visual analysis in unconstrained videos: we generate a middle-level atomic representation to fuse visual and audio signals and do not rely on precise object extraction. Experiments over the challenging Kodak's consumer benchmark videos demonstrate the effectiveness.

REFERENCES

ANEMUELLER, J., BACH, J., CAPUTO, B., ET AL. 2008. Biologically motivated audio-visual cue integration for object categorization. In *Proceedings of the International Conference on Computational Science*.

BARZELAY, Z. AND SCHECHNER, Y. 2007. Harmony in motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

BEAL, M. AND JOJIC, N. 2003. A graphical model for audiovisual object tracking. *IEEE Trans. Patt. Anal. Mach. Intell. 25*, 7, 828–836.

BIRCHFELD, S. 2007. Kit: An implementation of the Kanade-Lucas-Tomasi feature tracker. http://vision.stanford.eduj/∼birch.

CHANG, S., ELLIS, D., JIANG, W., ET AL. 2007. Large-scale multimodal semantic concept detection for consumer video. In *Proceedings of the ACM SIGMM Workshop on Multimedia Information Retrieval*.

CHEN, Y. AND WANG, J. 2004. Image categorization by learning and reasoning with regions. *J. Mach. Learn. Resear. 5*, 913–939.

CHU, S. AND NARAYANAN, S. 2008. Environmental sound recognition using mp-based features. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, 1–4.

CRISTANI, M., MANUELE, B., AND MURINO, V. 2007. Audio-visual event recognition m surveillance video sequences. *IEEE Trans. Multimedia 9*, 2, 257–267.

DEMENTHON, D. AND DOERMANN, D. 2003. Video retrieval using spatial-temporal descriptors. In *Proceedings of ACM Multimedia*, 508–517.

DENG, Y. AND MANJUNATH, B. 2001. Unsupervised segmentation of color-texture regions in images and video. *IEEE Trans. Patt. Anal. Mach. Intell. 23*, 8, 800–810.

FRIEDMAN, J., HASTIE, T., AND TIBSHIRANI, R. 2000. Additive logistic regression: a statistical view of boosting. *Ann. Stat. 28*, 22, 337–407.

GALMAR, E. AND HUET, B. 2007. Analysis of vector space model and spatiotemporal segmentation for video indexing and retrieval. In *Proceedings of the ACM International Conference on Image and Video Retrieval*.

GRAUMAN, K. AND DARREL, T. 2005. The pyramid match kernel: Discriminative classification with sets of image features. In *Proceedings of the IEEE International Conference on Computer Vision*. 1458–1465.

IWANO, K., YOSHINAGA, T., TAMURA, S., AND FURUI, S. 2007. Audiovisual speech recognition using lip information extracted from side-face images. *EURASIP J. Audio Speech Music Process. 1*, 4–4.

JEPSON, A., FLEET, D., AND EL-MARAGHI, T. 2003. Robust online appearance models for visual tracking. *IEEE Trans. Patt. Anal. Mach. Intell. 25*, 10, 1296–1311.

JIANG, W. 2010. Advanced techniques for semantic concept detection in unconstrained videos. Ph.D. thesis, Columbia University.

KAUCIC, R., DALTON, B., AND BLAKE, A. 1996. Real-time lip tracking for audio-visual speech recognition applications. In *Proceedings of the 2nd European Conference on Computer Vision*. 376–387.

KRSTULOYIC, S. AND GRIGONYAL, R. 2006. MPTK Matching Pursuit made tractable. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*. 496–499.

LOUI, A., LUO, J., AND CHANG, S. 2007. Kodak's consumer video benchmark data set: concept definition and annotation. In *Proceedings of the ACM SIGMM Workshop on Multimedia Information Retrieval*. 245–254.

LOWE, D. 2004. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision 60*, 2, 91–110.

LUCAS, B. AND KANADE, T. 1981. An iterative image registration technique with an application to stereo vision. In *Proceedings of the Imaging Understanding Workshop*. 121–130.

MALLAT, S. AND ZHANG, Z. 1993. Matching pursuits with time-frequency dictionaries. *IEEE Trans. Sign. Process. 41*, 12, 3397–3415.

MARON, O. AND LOZANO-PEREZ, T. 1998. A framework for multiple-instance learning. In *Proceedings of the Conference on Advances in Neural Information Processing Systems*. 570–576.

NIEBLES, J., HAN, B., FERENCZ, A., AND LI, F. 2008. Extracting moving people from internet videos. *Int. J. Comput. Vision 79*, 3, 299–318.

OGLE, J. AND ELLIS, D. 2007. Fingerprinting to identify repeated sound events in long-duration personal audio recordings. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*. 233–236.

PETITCOLAS, F. 2003. Mpeg for matlab. http://www.petitcolas.net/fabien/software.mpeg.

SHI, J. AND TOMASI, C. 1994. Good features to track. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 593–600.

SMEATON, A. F., OVER, P., AND KRAAIJ, W. 2006. Evaluation campaigns and TRECVid. In *Proceedings of the ACM SIGMM Workshop on Multimedia Information Retrieval*. 321–330.

STAUFFER, C. AND GRIMSON, W. 2002. Learning patterns of activity using real-time tracking. *IEEE Trans. Patt. Anal. Mach. Intell. 22*, 8, 747–757.

TIEU, K. AND VIOLA, P. 2000. Boosting Image retrieval. *Int. J. Comput. Vision 56*, 1–2, 228–235.

YANAGAWA, A., HSU, W., AND CHANG, S. 2006. Brief descriptions of visual features for baseline TRECVID concept detectors. Columbia University ADVENT Tech. rep. 219-2006-5.

YANG, C., DONG, M., AND HUA, J. 2006. Region-based image annotation using asymmetrical support vector machine-based multiple-instance learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2057–2063.

ZHOU, H., YUAN, Y., AND SHI, C. 2009. Object tracking using SIFT features and Mean Shift. *Comput. Vis. Image Understand. 113*, 3, 345–352.

Received March 2010; revised April 2010, May 2010; accepted May 2010