Semantic Learning for Audio Applications: A Computer Vision Approach

Rahul Sukthankar[1,2] Yan Ke[2] Derek Hoiem[2]
rahuls@cs.cmu.edu yke@cs.cmu.edu dhoiem@cs.cmu.edu

[1]Intel Research Pittsburgh [2]School of Computer Science, Carnegie Mellon

Abstract

Recent work in machine learning has significantly benefited semantic extraction tasks in computer vision, particularly for object recognition and image retrieval. We argue that the computer vision techniques that have been successfully applied in those settings can effectively be translated to other domains, such as audio. This claim is supported by recent results in music vs. speech classification, structure from sound, robust music identification and sound object recognition. This paper focuses on two such audio applications and demonstrates how ideas from computer vision map naturally to these problems.

1. Introduction

Computer vision research may initially appear to have little to contribute to problems in the audio domain. Audio requires processing of 1-D signals through time while computer vision traditionally focuses on the interpretation of one or more 2-D images. This paper argues that many problems in the audio domain transform naturally into a form that can be effectively tackled by computer vision techniques. This belief is motivated by the observation that audio researchers commonly employ 2-D time-frequency representations, such as spectrograms, when analyzing sound or speech. Currently, these representations are treated as images only for purposes of visualization. Our idea is to apply recent computer vision techniques, such as boosted classifiers [36], scanning windowed object detectors [28], local-descriptor based object recognition [21] and sub-image retrieval [18, 25] to these "images". The common thread behind all of these computer vision approaches is that they employ machine learning to extract semantics from the image and utilize statistical methods to robustly cope with noisy matches. This paper examines the merits of such approaches in the context of several real-world audio applications.

The remainder of the paper is organized as follows. Section 2 reviews the related work on semantic learning in audio, focusing particularly on recent efforts that appear to be inspired by computer vision approaches. Section 3 describes our work in music identification from noisy audio snippets, derived from techniques for 2-D sub-image retrieval. Section 4 summarizes our research in sound object detection, framed as a classical 2-D object detection problem. Section 5 concludes the paper.

2. Related Work

Much of the computer vision research relevant to semantic learning in audio focuses on the problem of audio-visual fusion. Hershey and Movellan [12] developed parametric models for the joint distribution between video and audio data, enabling them to highlight image regions that correspond to a particular sound. Fisher *et al.* [8] present a nonparametric approach to the same problem that exploits mutual information between the signals. More recently, Kidron *et al.* [19] propose an approach based on linear programming (termed "canonical correlation analysis") that associates pixels in a video with particular sounds in the audio stream. However, the focus of this paper is on applying computer vision techniques directly to the audio data and is less concerned with audio-visual fusion.

Research in auditory scene analysis [4] develops parallels between scene analysis in computer vision and the perceptual organization of sound. Structure from sound (SFS) [34] explores the problem of simultaneously recovering the locations of a set of microphones and a set of sound events that occur at arbitrary locations and unknown times in the environment. SFS exploits the analogy between this problem and a well-studied problem in computer vision: structure from motion. The goal in structure from motion (SFM) is to simultaneously recover the scene geometry (3D positions of a set of point features) and the camera pose at which each image was acquired. The sensor model for the two problems is quite different: in SFM, the cameras observe the relative angle for each

Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)
0-7695-2646-2/06 $20.00 © 2006 IEEE

feature point (since each location maps to a 3D ray), while in SFS, the microphones measure time-of-arrival (which corresponds to a relative range). However, the two problems can still be formulated in a similar manner, as a least squares problem over the extrinsic calibration parameters of the sensors and the global locations (in space and time) of the observed events. In its most general form, solving this optimization problem can be very challenging. The popular "factorization method" [35] approach for SFM assumes orthographic projection, where the rays from the feature points to each camera are assumed to be parallel (i.e., that the observed scene is far away from each camera). This results in an affine structure that can be efficiently solved using SVD. An analogous simplification is employed in [34] for SFS; the sound events are assumed to be sufficiently distant from each microphone such that their incident rays at each microphone are parallel. Just as the orthographic projection ignores perspective effects in computer vision, Thrun's "orthocoustic" model ignores "perspective" for an auditory scene. The success of SFS on both simulated and real-data supports our belief that ideas from computer vision could benefit problems in audio.

Several researchers have recently proposed the idea of extracting semantics from spectrograms using image processing techniques. Most of these approaches advocate manually-engineered descriptors, such as Haitsma and Kalker's music recognition system [11]. In their approach, the output of a single Haar-like feature (scanned at specified locations over the spectrogram) is used as a song signature that can be recognized in the presence of noise. While such techniques can achieve reasonable results, our experiments demonstrate that descriptors trained using real data generally outperform engineered representations.

Casagrande *et al.* [6] address the problem of distinguishing music from speech in audio streams using an approach that is also motivated by the Viola-Jones face detector architecture [36]. They employ AdaBoost [9] in conjunction with Haar-like wavelet features [26] to learn a windowed classifier that can be efficiently scanned over the spectrogram "image". Casagrande *et al.* report that this approach significantly outperforms the best earlier results [30] on the same dataset. This lends further support to our claim that audio problems can often easily be transformed into computer vision problems for which good solutions have already been developed.

There is some similarity between music vs. speech recognition and music identification. The former is analogous to the computer vision problem of detecting a known object (e.g., a human face) in an image; the latter is analogous to sub-image retrieval, where the goal is to find the best match for a partial noisy query. The former can be tackled using a single binary classifier, whereas the latter is a multi-class problem with thousands of classes, none of which is known at training time. Thus, an important challenge in music identification is to develop a discriminative yet general representation for audio that can generalize to unknown songs. Section 3 describes our approach to music identification, and additional details are given in [16].

There has been some research in the audio community regarding detection of auditory objects in spectrogram images. Smaragdis [33] employed non-negative factorization (NMF) to spectrograms to discover auditory objects in audio scenes. The work appears to have been motivated by observations that NMF, on 2-D images (such as human faces) often recovers sparse, parts-based representations [20]. Section 4 describes how we frame sound object detection as a classical 2-D object detection problem to recognize sounds in audio streams; additional details are available in [13].

3. Music Identification as Sub-image Retrieval

The goal of music identification is to reliably recognize a song from a small sample of noisy audio. For instance, a user who wants to know the name of a song playing at a party could send a few seconds of audio using her mobile phone to a music identification server and receive a text message with the title of the song. This task is challenging for several reasons. First, the query can be significantly corrupted by the distortions induced by typical portable recording devices or due to noise from ambient sounds. Second, the audio sample from the query will typically match only a small portion of the target song, such that a traditional digital signature computed over the query is unlikely to match the signature of the entire song. Third, a practical music identification system should scale, both in accuracy and speed, to databases containing hundreds of thousands of songs. Finally, the system's representation must be able to handle a growing database of songs; for a learning system, the challenge is that the songs in the training set are not the same as the ones in the testing set — in other words, music recognition is a multi-class classification problem where the classes are not specified until the testing phase. Recently, the music identification problem has attracted considerable attention, both from commercial companies [1–3] and researchers [5, 11]. However, the task remains challenging, particularly for noisy real-world queries.

Our approach casts music identification into the framework of sub-image retrieval. In sub-image retrieval, the goal is to find the best match between a partial query image and images stored in a database. The query image is typically a transformed version of the original (e.g., cropped, scaled, rotated) and often corrupted by noise (e.g., illumination effects or encoding artifacts). One successful approach [18] to sub-image retrieval relies

on local descriptors [17, 22] to identify likely candidate matches at a patch level and verifies the hypotheses at a subsequent stage using RANSAC [7]. The benefits of the approach are that local descriptors are robust to cropping and occlusion since many patches remain unaffected by such transformations, whereas a global descriptor would be severely impacted. In music identification, the noisy sample of audio corresponds to the query sub-image and the database of clean song recordings corresponds to the database of complete images. Under this analogy, the query audio snippet is a "cropped" version of one of the originals, and portions of the query could be "occluded" by loud sounds or corrupted by encoding/recording noise. However, it is important to understand that the analogy alone does not solve the music identification problem; rather, it enables researchers to identify a series of important questions:

1. What is the class of "geometric" transformations to which our representation must be robust?
2. Are there good interest point detectors for spectrogram images?
3. What are appropriate local descriptors for spectrogram images?
4. Is there an appropriate analog for "geometric verification" in music identification?

We briefly examine each of these in turn. First, although spectrograms are 2D images, they do not exhibit all of the variations of camera-generated natural scene images. For instance, rotation- and scale-invariance is not required (nor desirable) in spectrogram matching. Similarly, invariance to illumination is not an issue. On the other hand, spectrograms will typically contain superpositions of the various sound sources in the environment (analogous to a visual scene containing many translucent and transparent objects). Second, while it would be intellectually-interesting to find good "SIFT-like" interest point detectors for audio, the need for locating keypoints in audio is less important since a dense, scanning search is more feasible in the absence of rotation and scaling. Third, it is clear that one cannot blindly apply existing descriptors from computer vision, such as PCA-SIFT. Proposed descriptors such as Haitsma and Kalker's corner features [11] are the first step in the right direction. However, we strongly believe (and demonstrate) that a better approach is to learn the right descriptor from real data using machine learning (as detailed below). Finally, the idea of RANSAC-based "geometric verification" in sub-image retrieval translates well to the music identification problem: one can verify that several local matches between a query and a candidate song are "geometrically consistent" in terms of temporal offset.

3.1. Local Descriptors for Audio

Given a short segment of distorted audio data, we would like the music identification system to quickly find the matching segment of undistorted audio in a large database. The system should meet the following performance requirements: high recall, high precision, query using short audio clips, and fast retrieval. To achieve high recall, its representation must be sufficiently descriptive to distinguish between similar-sounding songs. High recall, on the other hand, demands that the representation also be highly resistant to distortions caused by background noise or poor recording quality. For instance, a song played over low-quality speakers and recorded using a laptop's built-in microphone will sound significantly different from the same song played over high-fidelity speakers and recorded using a professional microphone. Since we want the ability to identify a song based on only a few seconds of audio sampled at an arbitrary point in the song, the representation should be local and robust to small shifts in time. Furthermore, a music identification system should scale to large music databases, returning accurate responses in a few seconds on queries against hundreds of thousands of songs. This scaling requirement indicates that the representation should be computationally inexpensive and efficiently indexable.

Creating a feature representation that meets all of these criteria is a challenging task. The raw representation of an audio signal (as amplitude vs. time) is extremely sensitive to small distortions and perceptual information is difficult to extract directly (Figure 1a). The spectrogram representation is computed using the short-term Fourier transform and represents the power contained in contained in 33 logarithmically-spaced frequency bands, measured over 0.372 s windows in 11.6 ms increments (our spectrogram images are generated using the parameters given by Haitsma and Kalker [11]). In the spectrogram image (Figure 1b), corrupted audio bears some visual similarity to its original but the signal differences due to different audio sources are also highly visible. Although the process of converting the time-domain signal into a spectrogram image illuminates important similarities and differences in the audio, simply comparing spectrograms using correlation would be inaccurate and slow. Instead, we advocate learning a small set of filters whose responses are robust to expected distortions while preserving the information needed to distinguish between different songs (Figure 1c). Rather than attempting to manually engineer such a suitable set of filters, we define a broad class of candidate filters and apply machine learning techniques to identify a small subset that performs well together. To determine an appropriate family of filters for this task, it is helpful to examine the characteristics of spectrogram images that are distinctive (sensitive to the particular song) while being resistant to expected distortions. These characteristics include: (a) differences of power in neighboring frequency bands at a particular time; (b) differences of power across time

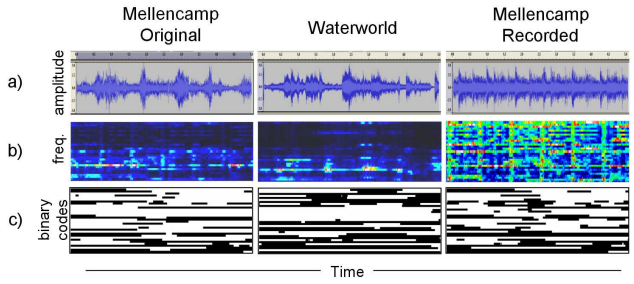

Figure 1. Representing audio. Three 10-second snippets of audio are shown: John Mellencamp original, Waterworld soundtrack, and John Mellencamp recorded. It is difficult to determine which snippet matches the song in the waveform audio representation (a). In the spectrogram images (b), certain similarities between the two Mellencamp snippets and distinguishing differences between the Mellencamp and Waterworld snippets become noticeable. Matching snippets are easily identified using our learned descriptions (c).

Figure 2. Candidate filter set. We select a compact set of filters from the filter class employed for object detection by Viola and Jones. When applied to spectrogram images, these filters capture important time-frequency characteristics of audio.

within a particular frequency band; (c) shifts in dominant frequency over time; (d) peaks of power across frequencies at a particular time; and (e) peaks of power across time within a particular frequency band. The proposed filter family should be able to capture these aspects while operating along different frequency bands with different bandwidths and over different extents of time. Filters with large bandwidths and time-widths are more robust to certain distortions, but filters with short bandwidths and time-widths can capture discriminative information that the former filters cannot. If we view the spectrogram as a simple 2-D grayscale image, we can see that the class of Haar wavelet-like filters [26, 36] meets these requirements (Figure 2).

In our system, each filter type can vary in band location from 1 to 33, in bandwidth from 1 to 33, and in time from 1 frame (11.6 ms) to 82 frames (951 ms) in exponential steps of 1.5, resulting in a set of roughly 25,000 candidate filters. From this large candidate set, we select M discriminative filters and corresponding thresholds to generate an M-bit vector that represents overlapping segments of audio (Figure 1c).[1] This vector, termed the descriptor, can be quickly computed using integral images [36] and is sufficiently stable across distortions to enable retrieval by direct hashing in the database. We describe how to learn the description below.

Of course, a single descriptor cannot contain enough information to accurately identify the song matching the given query from among hundreds of thousands of full-length songs. To represent several seconds of an audio snippet, we compute descriptors for overlapping windows of audio every 11.6 ms. Thus, for a ten-second snippet of audio, our signature consists of 860 descriptors. This signature is the basis for matching and retrieval.

3.2. Filter Selection and Modeling

The previous subsection described how we can treat the time-frequency representation of an audio signal as an image and outlined the set of candidate filters that operate on the spectrogram image. This subsection details our method for selecting a subset of those filters (and corresponding thresholds) to create a compact representation for each local region of the spectrogram image. The goal is to build a representation in which an original audio segment and its distorted versions will generate highly-similar descriptors, while audio segments from two different songs will generate dissimilar descriptors.

The descriptors capture only the *local* similarity between a pair of short segments of audio. To correctly evaluate the match between the query snippet and a song in the database, we need to compute the probability that an entire signature (the series of descriptors computed on overlapping audio windows) matches the other.

Additionally, we account for "occlusion" due to background noise that drowns out the signal or due to a poor mobile phone connection. We assume that each descriptor in the signature was generated either by the original song or by an occluding signal. We employ the Expectation Maximization (EM) algorithm [24] and a simple dependency model to automatically determine whether a given descriptor in a sequence corresponds to the song or an occlusion and to compute the likelihood that one signature matches another (see **??** for additional details).

3.3. Learning Compact Audio Descriptions

Our goal is to build a description that enables us to determine the probability that two (potentially distorted) audio snippets were both sampled from the same position of the same song. Formally, this entails learning a classifier $H(x_1, x_2) \rightarrow y$=$\{-1, 1\}$, where x_1 and x_2 are two spectrogram images and the label y denotes whether the images derive from the same original audio source (y=1) or not (y=-1). One popular method of building a description for object recognition is to define a large class of filters and use AdaBoost [9, 29] to select a small subset of those filters for classification. We apply a novel pairwise variant of this method. Our classifier is an ensemble of M weak

[1] Our current implementation uses $M = 32$.

classifiers, $h_m(x_1, x_2)$, each with an associated confidence, c_m. Our weak classifiers are composed of a filter f_m and a threshold t_m, such that $h_m(x_1, x_2)$=$sgn[(f_m(x_1) - t_m)(f_m(x_2) - t_m)]$. In other words, if two examples generate filter response values on the same side of the threshold, they are labeled by the weak classifier as deriving from the same portion of audio; otherwise, they are labeled as coming from different audio snippets. Note that this formulation differs from standard AdaBoost in that labels are assigned to *pairs* of filter responses. Once the weak classifiers are learned, any spectrogram x can be transformed into an M-bit vector, allowing fast indexing through hashing techniques.

One way to learn these weak classifiers would be through the standard AdaBoost framework in which, iteratively, weak classifiers are learned and all of the data is re-weighted. Such an approach, however, would produce poor results in this case for the following reason: no weak classifier can perform better than chance, on average, on the *non-matching* example pairs! This may seem an odd assertion, but the proof is summarized as follows. Suppose we have x randomly drawn from distribution D, a filter f_m, and a threshold t_m, such that $P(f_m(x) < t_m) = p$, with $0 \leq p \leq 1$. If we independently and randomly draw two non-matching examples x_1 and x_2 from D, then the probability that x_1 and x_2 fall on different sides of t_m is given by

$$P\left(h_m(x_1, x_2) = -1\right) = 2p(1-p) \leq 0.5. \qquad (1)$$

Thus, a pair of non-matching (y=-1) examples will incorrectly be classified as matching at least half of the time for a sufficiently large sample size, violating the weak classifier condition of AdaBoost. We resolve this issue by employing an asymmetric pairwise boosting algorithm, in which only the matching (y=1) pairs are re-weighted and the weights of matching pairs and non-matching pairs are normalized such that the sum of each is equal to one-half. Our algorithm is detailed in Figure 3.

From Equation 1, we also note that we can explicitly calculate the probability of error for non-matching pairs for a particular filter and threshold if we know the distribution of the filter responses. We observe that this distribution can be estimated from the single members of the matching pairs, providing two results: (1) the median is the optimal threshold for non-matching pairs; and (2) when the filters are loosely correlated, we do not need non-matching pairs — providing a two-fold speed-up in training or the ability to employ a larger training set of matched pairs at no additional computational cost. Experiments reveal that all thresholds learned by the pairwise boosting are approximately at the median of the filter response distribution and that approximating non-matching error in this manner has minimal impact on classification accuracy.

Pairwise Boosting

input: sequence of n examples $\langle(x_{11}, x_{21})\rangle..\langle(x_{1n}, x_{2n})\rangle$, each with label $y_i \in \{-1, 1\}$

initialize: $w_i = \frac{1}{n}, i = 1..n$

for m = 1..M

1. find the hypothesis $h_m(x_1, x_2)$ that minimizes weighted error over distribution w, where $h_m(x_1, x_2) = sgn[(f_m(x_1) - t_m)(f_m(x_2) - t_m)]$ for filter f_m and threshold t_m
2. calculate weighted error: $err_m = \sum_{i=1}^{n} w_i \cdot \delta(h_m(x_{1i}, x_{2i}) \neq y_i)$
3. assign confidence to h_m: $c_m = \log(\frac{1-err_m}{err_m})$
4. update weights for matching pairs: if $y_i = 1$ and $h_m(x_{1i}, x_{2i}) \neq y_i$, then $w_i \leftarrow w_i \cdot exp[c_m]$
5. normalize weights such that $\sum_{i:y_i=-1}^{n} w_i = \sum_{i:y_i=1}^{n} w_i = \frac{1}{2}$.

final hypothesis:

$H(x_1, x_2) = sgn(\sum_{m=1}^{M} c_m h_m(x_1, x_2))$

Figure 3. Summary of our pairwise boosting algorithm for learning a hypothesis that can determine whether the members of a pair, x_1 and x_2, belong to the same class (match) or belong to different classes. Note that the algorithm is asymmetric in that only the matching example pairs are boosted. This is necessary because our simple classifiers cannot achieve better accuracy than chance on the non-matching pairs and, thus, fail to meet the AdaBoost weak classifier criterion.

We note that several other researchers have proposed related pairwise methods for pose estimation, face recognition, and object recognition. Shakhnarovich *et al.* [32] independently select the filters that most preserve similarity. Ren *et al.* [27] learn features for identifying human motion from silhouette images using this technique. Jones and Viola [15] select a set of filters using AdaBoost, with weak classifiers based on thresholding the difference of responses for same-face and different-face pairs. Mahamud and Hebert [23] model the distance between two data points as the probability that the points have different labels and estimate that probability.

Our learned set of filters (M=32 in our implementation) greatly improves upon the descriptors recently developed by Haitsma and Kalker for music identification [11]. The Haitsma-Kalker filters compute the difference between neighboring frequencies at neighboring times. These filters are equivalent to the diagonal Viola-Jones filters (Figure 2c) with a bandwidth of 2 bands and a time-width of 2 frames. After learning our description, we noticed several commonalities among the filters. One is that the time-widths tend to be large (usually 54 frames or longer out of a maximum of 82 frames). Filters that have a smaller time-width tend to have a large band-width. These characteristics

support our belief that filters that have a large extent in a particular dimension can "average out" much of the noise and distortion induced by poor-quality recordings. We also noticed that, out of the 32 filters, 31 either measure the difference in two sets of frequency bands at a particular time interval or a peak across frequency bands at a particular time interval. Thus, the learned filters are highly robust to noise that affect all bands intermittently but are more susceptible to distortions that affect a particular frequency range over long durations.

3.4. Retrieval

Using the representation described in the previous sections, we build signatures for all of the songs in the database. During retrieval, we perform a similarity search for each of the query snippet's descriptors against this signature database. The large size of our database and the high number of queries required for each snippet motivates us to seek efficient schemes for similarity search in high-dimensional (typically 32-bit) descriptor space. A natural choice is locality-sensitive hashing (LSH) [14], a technique that enables approximate similarity searches in sub-linear time, particularly since it is so well-suited for the Hamming distance metric [10]. Our initial experiments using LSH gave excellent results, but, somewhat surprisingly, we discovered that our descriptors are so robust that *direct indexing*, using a classical hash table, greatly reduced running time without significantly impacting accuracy. We describe this indexing approach in the remainder of this section.

We hash all of the signatures into a standard hash table (keyed by appropriate M-bit descriptors). We define those descriptors within a Hamming distance of 2 from the given query to be near-neighbors. These are retrieved with the following sequence of exhaustive probes. First, we probe the hashtable with the query; this retrieves all matches within a Hamming distance of 0. Next, we make M additional probes in the hash, each consisting of the query descriptor with a single bit flipped; this finds matches at a Hamming distance of 1. Finally, we repeat this process with every combination of two-bit flips to retrieve those descriptors at a Hamming distance of 2. While such an approach may initially appear to be inefficient, we have observed that it is significantly faster than LSH for our application because each probe is so inexpensive and it returns exact rather than approximate results. We have observed that the use of unweighted Hamming distance instead of classifier confidence as a basis for descriptor similarity is a reasonable approximation, since we found the confidence values for different weak classifiers to be nearly equal in our experiments.

Once all of the near neighbors have been found, we need to identify the song that best matches the set of descriptors in the query. Rather than simply voting based on the number of matches, we employ a form of geometric verification that is similar to that used in object recognition using local features [21]. For each candidate song, we determine whether the matched descriptors are consistent over time. For this, we use RANSAC [7] to iterate through candidate time alignments and use the EM score, the likelihood of the query signature being generated by the same original audio as the candidate signature, as the distance metric. We have explored two alignment models. The first assumes that the query can be aligned to the original once a single parameter (temporal offset) has been determined. In this case, the minimal set is a single pair of matching descriptors. The second model assumes that the query could be a temporally scaled (linearly stretched or compressed) version of the original. This model is defined by two parameters (speed ratio and offset) and requires a minimal set of two matching descriptors. More complicated temporal distortion models are certainly possible. In practice, we have found that the first model gives accurate results, particularly since our query snippets are short. We find that RANSAC converges in fewer than 500 iterations even in the presence of significant occlusion. Once all of the retrieved candidates have been aligned, we select the song with the best EM score, assuming that it passes a minimum threshold.

3.5. Experimental Results

This paper summarizes some key experimental results. Additional experiments on descriptor-level and song-level accuracy, as well as the impact of different parameter settings are reported in [16].

We need to train the two parts of our system: the filters for extracting descriptors and the EM noise model. Both requires training data consisting of aligned pairs of filter outputs. This poses a chicken-and-egg alignment problem: how can we accurately align noisy recordings to original songs before learning good descriptions? Our solution is to bootstrap the learning process with synthetically-distorted songs, for which the alignment is known. From these we learn a set of filters that, while insufficiently accurate for music identification in noisy environments, is suitable for training data alignment.

The training data consists of 78 songs played through low-quality speakers and recorded using low-quality microphones, aligned to the originals using the bootstrap filters. We learn the 32-bit filters and the EM noise parameters as described above. Next, we record test data in a completely different environment using different microphones, speakers, computers and recording rooms. The experiments use two challenging real-world test sets designed to exemplify worst-case scenarios. The first consists of 71 songs played at a low volume and recorded with a distorted microphone (denoted as "Test A"). The

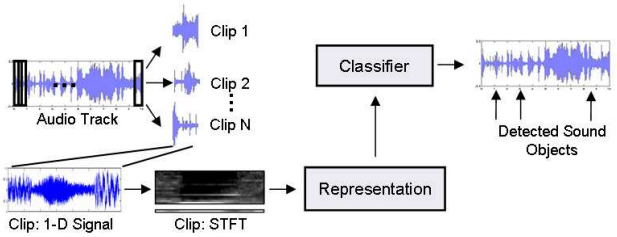

Figure 4. To detect sound objects, we follow a similar approach to the windowed object detection approach, with a representation suited for detecting sounds in the audio domain.

second more difficult set contains 220 songs captured with a very noisy recording setup (denoted as "Test B"). In many cases, the noise drowns out the music to such a degree that the song is barely audible to humans. These recordings are drawn from a database of 145 albums with 1862 songs spanning a large variety of music genres including classical, vocal, rock and pop. Since each query could match up to 1861 false positives, the baseline accuracy of the tasks is only 0.05%. On Test A, our system achieves a precision of over 97% at a recall of 90%, and on Test B, a precision of 93% for a recall of 80%. These results are dramatically better than those of [11], and confirm our hypothesis that learning descriptors from real data can significantly improve recognition accuracy in this domain.

4. Detecting Sound Objects

In sound detection, the goal is to identify particular classes of sounds, such as dog barks, gunshots, screams, laughing, and speech. Often, these sounds often have a finite extent in the audio stream, and, similarly to detecting visual objects, the task becomes one of segmenting the objects of interest from the remaining signal and classifying them. In visual detection, the segmentation step is often bypassed by performing a windowed search over locations and scales [28, 31, 36], which provides an over-complete segmentation of the image that is effective when the object has a fixed shape (e.g., human faces or cars). For sound object detection using SOLAR [13], we use the same general approach, segmenting the audio stream into overlapping windows and classifying each window into the object of interest or background.

Figure 4 illustrates our algorithm. We compute a short-time Fourier transform (STFT) of the audio signal and divide it into overlapping, equal-length windows in time. The length of time is set to be the median duration of the training sound objects, and the overlap is one-eighth of the duration. Because the "scale" of audio (the duration) is meaningful, we do not search over scale in our audio stream. We represent each time-window of the STFT, first decomposing it into an average total power and the percent of power in each frequency band. We compute 138 statistics that measure mean and variance of the power percentile in each frequency channel and of the total power, bandwidth, the most powerful frequency channel, the number of peaks in power over time, the regularity of power peaks, the range of the total power over time, and time-localized frequency percentiles over various frequency ranges. Analogous to the manner in which computer vision researchers develop an independent detector for each object of interest (such as faces, cars, pedestrians), SOLAR learns a binary classifier for each sound object from training data. SOLAR detectors use boosted decision trees [29] to perform the binary classification. Each tree makes decisions based on a discriminative subset of the statistics and outputs a confidence based on the class-conditional log likelihood ratio at each leaf node.

Unlike traditional object detection, where thousands of positive examples are used to learn each object category, our goal is to build useful sound object detectors using a small number of examples. We have trained and tested SOLAR using sound clips obtained from `www.findsounds.com` for the following sound classes: car horn, close door, dog bark, door bell, explosion, gunshot, laser gun, light saber, male laugh, meow, scream, and sword clash. Positive test examples included these sounds mixed with background sounds randomly selected from audio streams from movies. The background class consisted of the audio streams from several movies that did not contain the sound object of interest. At false positive rates of 10 FP/hr, 50 FP/hr, and 100 FP/hr, we obtain average detection rates of 37%, 60%, and 72%, respectively. The highest accuracy was obtained for door bells, meows, and phone rings, which may be attributed to the sounds' high pitch or regularity. The most difficult classes were gunshots, explosions, door closings, and male laughs, likely due to the low pitch that blends with the background noise, and, in the case of male laughs, confusion with voices. Further details of our approach are available in [13] and at `www.cs.cmu.edu/~dhoiem/projects/solar/`.

5. Conclusion

Extracting semantics from audio data is a challenging and active research area. We observe that many tasks in audio transform naturally to problems that can be effectively addressed using computer vision techniques. This observation is supported by recent work in structure from sound, music vs. speech discrimination and audio-visual fusion. This paper describes our contributions to the field, specifically in the areas of music identification and sound object detection. We explore the analogy between music identification and 2-D sub-image retrieval and show that a local descriptor based approach significantly outperforms current approaches in content-based music

identification on real-world data. We frame sound object detection in the context of a scanning windowed binary classifier for each "object" of interest that can detect the given sound under noisy conditions. In both of our systems, machine learning plays a pivotal role in finding the right representation for audio. For music identification, we use a novel pairwise variant of boosting to learn a generic discriminative descriptor for music that can tackle a multi-class classification problem with thousands of classes. For sound object detection, we build boosted binary classifiers that can recognize the target sound object with only a small amount of training data. We believe that computer vision ideas have immediate applicability and direct relevance to semantic extraction in many domains and we hope that this paper encourages computer vision researchers to explore such opportunities.

References

[1] AT&T Wireless. `http://www.attwireless.com/`.

[2] Musikube. `http://www.musikube.com/`.

[3] Shazam Entertainment. `http://www.shazam.com/`.

[4] A. Bregman. *Auditory Scene Analysis: The Perceptual Organization of Sound*. MIT Press, 1990.

[5] P. Cano, E. Batlle, T. Kalker, and J. Haitsma. A review of algorithms for audio fingerprinting. In *Workshop on Multimedia Signal Processing*, 2002.

[6] N. Casagrande, D. Eck, and B. Kégl. Frame-level speech/music discrimination using AdaBoost. In *Proc. International Symposium on Music Information Retrieval*, 2005.

[7] M. Fischler and R. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6), 1981.

[8] J. Fisher III, T. Darrell, W. Freeman, and P. Viola. Learning joint statistical models for audio-visual fusion and segregation. In *Proc. NIPS*, 2000.

[9] Y. Freund and R. Schapire. Experiments with a new boosting algorithm. In *Proc. Intl. Conf. on Machine Learning*, 1996.

[10] A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. In *Proc. Intl. Conf. on Very Large Databases*, 1999.

[11] J. Haitsma and T. Kalker. A highly robust audio fingerprinting system. In *Proc. Intl. Conf. on Music Information Retrieval*, 2002.

[12] J. Hershey and J. Movellan. Using audio-visual synchrony to locate sounds. In *Proc. NIPS*, 1999.

[13] D. Hoiem, Y. Ke, and R. Sukthankar. SOLAR: Sound object localization and retrieval in complex audio environments. In *Proc. Intl. Conf. on Acoustics, Speech and Signal Processing*, 2005.

[14] P. Indyk and R. Motwani. Approximate nearest neighbor – towards removing the curse of dimensionality. In *Proc. Symposium on Theory of Computing*, 1998.

[15] M. Jones and P. Viola. Face recognition using boosted local features. Technical Report MERL-TR-2003-25, Mitsubishi Electric Research Laboratory, 2003.

[16] Y. Ke, D. Hoiem, and R. Sukthankar. Computer vision for music identification. In *Proc. Computer Vision and Pattern Recognition*, 2005.

[17] Y. Ke and R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In *Proc. Computer Vision and Pattern Recognition*, 2004.

[18] Y. Ke, R. Sukthankar, and L. Huston. Efficient near-duplicate and sub-image retrieval. In *Proc. ACM Multimedia*, 2004.

[19] E. Kidron, Y. Schechner, and M. Elad. Pixels that sound. In *Proc. Computer Vision and Pattern Recognition*, 2005.

[20] D. Lee and H. Seung. Learning the parts of objects with non-negative matrix factorization. *Nature*, 401, 1999.

[21] D. Lowe. Object recognition from local scale-invariant features. In *Proc. Intl. Conf. on Computer Vision*, 1999.

[22] D. Lowe. Distinctive image features from scale-invariant keypoints. *Intl. Journal of Computer Vision*, 2004.

[23] S. Mahamud and M. Hebert. Minimum risk distance measure for object recognition. In *Proc. Intl. Conf. on Computer Vision*, 2003.

[24] G. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. Wiley Series in Probability and Statistics, 1997.

[25] Y. Meng, E. Chang, and B. Li. Enhancing DPF for near-replica image recognition. In *Proc. Computer Vision and Pattern Recognition*, 2003.

[26] C. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *Proc. Intl. Conf. on Computer Vision*, 1998.

[27] L. Ren, G. Shakhnarovich, J. Hodgins, P. Viola, and H. Pfister. Learning silhouette features for control of human motion. Technical Report CMU-CS-04-165, Carnegie Mellon University, 2004.

[28] H. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1), 1998.

[29] R. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3), 1999.

[30] E. Scheirer and M. Slaney. Construction and evaluation of a robust multifeature speech/music discriminator. In *Proc. Intl. Conf. on Acoustics, Speech and Signal Processing*, 1997.

[31] H. Schneiderman and T. Kanade. Object detection using the statistics of parts. *Intl. Journal of Computer Vision*, 2002.

[32] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter sensitive hashing. In *Proc. Intl. Conf. on Computer Vision*, 2003.

[33] P. Smaragdis. Discovering auditory objects through non-negativity constraints. In *Proc. Statistical and Perceptual Audio Processing*, 2004.

[34] S. Thrun. Affine structure from sound. In *Proc. NIPS*, 2005.

[35] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *Intl. Journal of Computer Vision*, 9(2), 1992.

[36] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. Computer Vision and Pattern Recognition*, 2001.