5

How to evaluate the sources in a soundscape?


5.1 Introduction

Humans can recognize events in the sonic environment (soundscape) seemingly effortlessly. However, when the details of the recognized sound are asked this performance is more questionable. As we will see in chapter 7 the agreement between two human annotators, both present during recordings, is low. Also, from my own anecdotal experience with annotating the amstel dataset from chapter 4 one keeps hearing new sounds and details, even after several serious attempts at a complete annotation over a two year period.

Despite the problems with human annotation it is the only feasible way of obtaining a ground truth for uncontrolled environments. Therefore, this chapter focuses on the development of a tool to facilitate real-world sound
Chapter 5. How to evaluate the sources in a soundscape?

annotation for training and benchmark purposes. It uses a set of simple algorithms to detect sonic events and to classify these events. The interaction between semantic content, in the form of annotations, and signal-based evidence forms the basis of future, more general, sound recognition systems. The annotation of everyday sounds must lead to an adequate description of the content of a sound-file in terms of the interval in which an event occurred. Annotation is a time-consuming, and knowledge intensive task, which is usually quite boring as well. This is probably the reason why there is currently only a single annotated database of sounds in realistic everyday conditions (van Grootel et al., 2009). Carefully selected everyday sounds in benign conditions have been used in other studies (Gygi et al., 2007; Marcell et al., 2000). However for these sounds the annotation problem is trivialized, because the datasets contain single sound events in a single file.

There are many difficulties associated with real-world sound annotation: The great within class diversity of sounds (e.g. cars at different distances and speeds) in combination with the co-occurrence of other classes makes it difficult to interpret a visual rendering of the signal as spectrogram and to annotate the visual representation without listening to the sounds in context. Visual inspection of spectro-temporal representations is an important aid for annotation, but attentive listening to the sound is essential. Sonic events are often difficult to recognize using sound as the only modality. It is important to annotate the sound during, or soon after, recording. The use of video information can be very helpful whenever the sound sources are clearly visible and easily attributable (which is often not the case). Anecdotic evidence suggests that annotation by someone who was not present when the sound was recorded is much more error-prone and often many sounds cannot be annotated in detail. For example, the difference between cars, truck, busses, and even motorcycles is usually not at all obvious.

The co-occurrence of multiple qualitatively different sonic events and sound producing processes can lead to very complex signals, e.g. coffee-making in a lively kitchen. In these cases it is difficult to track multiple uncorrelated processes and describe each in detail. One might aim to annotate the so-called foreground or, alternatively, the events that attract attention. However, this creates the new problem of determining what attracts attention or what to assign to the foreground. The large number of individually distinguishable events of a similar kind, such as singing birds in a forest, entails a lot of repetitive work. Realistic environments contain many barely audible events, e.g. distant speakers, which might or might not be included in the annotation. Not including these might unjustly punish a detection system that detects the valid, but un-annotated, events. Conversely, including even the faintest events is both time-consuming and prone to classification errors.

Finally, the determination of the precise moment of the start and end of audible events is subject to similar difficulties as those in the previous
point. Especially the detection of the on- or offset of a gradually developing
event, like a passing car in a complex environment, is often quite arbitrary.
If the measure of success of a recognition system is based on determining the
intervals in which events occur, the system is punished for any deviation of
this arbitrary choice. The difference between annotators who were present
and who were not, suggests that the sonic evidence may often be insufficient
(for the human listener). This poses a fundamental problem for each sound-
only annotation or recognition system, whether human or machine; a correct
recognition result may simply be impossible. Hence, a perfect ground-truth
is not a realistic goal for a real-world sound recognition system. Instead,
a performance equivalent to human performance when not present during
recording is more appropriate.

The current chapter focuses on an annotation tool that helps to provide
more insight in these problems and helps to alleviate a number of them. It
assists a human annotator by reducing the number of repetitive actions by
automatically suggesting annotations based on previous annotations. This
allows for the human annotator to accept the suggested annotation simply as
an instance of the proposed class, instead of having to select it from a (long)
list of possible classes. Within the annotation system we try to maximize
the probability that the true event class is on top of the list. Initially this list
is simply alphabetic. During manual annotation the class list is reordered
according to the estimated probability that a certain event is an instance of
the most likely classes.

In the next section, we will give an overview of the annotation system.
Furthermore, we present the data on which it is tested. In the third section
we will give the results of a pilot-experiment on a set of real-world recordings.
The chapter ends with a short discussion of the annotation process.

5.2 Methods

In this section we first describe the dataset that is used to test the annotation
system. This system is based on processing sound in the spectro-temporal
domain. Therefore, the sound signal is first pre-processed, which will be
explained in section 5.2.2. Subsequently, we describe how the sound is seg-
mented into regions that are likely to include the most energetic spectro-
temporal evidence of the main sources. In section 5.2.3 we show how these
regions are described in terms of a feature vector, and how this feature vec-
tor is used to classify the regions. The section is concluded with a system
overview, which is shown in figure 5.1
5.2.1 Dataset

The dataset was collected under different weather conditions on a number of days in March 2009 in the town of Assen (65,000 inhabitants, in the north of the Netherlands). The recordings were made by six groups of three students as part of a master course on sound recognition. Each group made recordings of three minutes at six different locations: a railway station platform, a pedestrian crossing with traffic lights, a small park-like square, a pedestrian shopping area, the edge of a forest near a cemetery, and a walk between two of the positions. Recordings were made using M-Audio Microtrack-II recorders with the supplied stereo microphone at 48 kHz and 24 bits stereo. This data, with annotations by the students, is available on http://daresounds.org.
5.2. Methods

5.2.2 Preprocessing

The first processing is the transformation of the audio signal to the time-frequency domain using the techniques described in section 2.3. From the resulting cochleogram the tone-fit and pulse-fit values as described in chapter 3 are calculated. The cochleogram is used in the next section for segmentation and the cochleogram, tone-fit and pulse-fit representation are used in the feature vectors as described in section 5.2.3.

Segmentation

The segmentation strategy is fairly basic. It is aimed at the inclusion of spectro-temporal maxima in the form of blobs in the spectral and/or temporal direction. These blobs become prominent by subtracting a strongly smoothed cochleogram from the original. The cochleogram is smoothed in the temporal direction through leaky integration with a time-constant $\tau = 5s$. The time constant $\tau$ determines the separation between fast, typically foreground, sonic events and slow, typically background, events. The leaky integration operation corresponds to a delay in the expression of mean energy values that is corrected by time-shifting the resulting values backwards with the time-constant. This time-shift leads to a delay equal to the time-constant, which is not problematic for off-line processing, but that is not desirable for online and real-time processing. The temporal smoothing of time-series $x(t)$ to yield $x_s(t)$ is defined by:

$$x_s(t) = x(t - \delta t)\exp(-\delta t/\tau) + x(t)(1 - \exp(-t/\tau)) \tag{5.1}$$

$\delta t$ denotes the frame step of $5ms$. In addition to temporal smoothing, the cochleogram is also smoothed in the frequency direction by taking a moving average over 7 channels. The difference between the original cochleogram and the smoothed cochleogram can be termed a fast-to-slow-ratio and is expressed in dB. The regions with a fast-to-slow-ratio of more than $2dB$ are assigned a unit value in a binary mask. This mask is smoothed with a moving average in both the temporal direction (25ms) and the spectral direction (5 channels). The final mask is obtained by selecting average mask values greater than 0.5, which smoothens region perimeters and reduces the number of supra-threshold time-frequency points in the inner-regions of the mask that lead to small holes in the mask. The final segmentation step is the estimation of individual coherent regions in the mask and to assign a unique number to each region. The smallest bounding box that contains the whole region is used to represent the region graphically (see figure 5.2). There are no special safeguards to ensure either that each region represents information of a single source, or that all information of the source is included in the regions. For example, when two cars pass at approximately...
the same time, a single region will represent both. Alternatively, sounds that are partially masked by (slowly developing) background sounds tend to break up into a number of smaller regions, that are each less characteristic of the source. Nevertheless, the current settings seem able to include important source information of a wide range of sources.

5.2.3 Feature vectors

The feature vectors must describe the source information represented by the regions. The 37-dimensional feature vector represents properties related to the physics of the source. Note that normal approaches to environmental sound feature estimation [Cowling and Sitte, 2003] make no effort to include source physics other than representing frequency content. The use of the $TF/PF$-values allows us to attribute signal energy to tonal, pulse-like, or noisy contributions, which result from either source limitations or transmission effects. Table 5.1 describes the feature vector. The feature vector reflects the channel contributions per region, the fast-to-slow ratio, and the distribution of tonal ($TF$) and pulse-like ($PF$) contributions. These signal descriptors are represented by 7 different percentile values from the histogram of the local indicators. Different percentile values might be indicative for different classes. For example, the 90 and 95 percentile values might be highly indicative for footsteps in noise, while the other percentiles might not discriminate from a the noisy contribution in a car passage.

Classification

Classification of regions based on the feature vector must lead to proposed classes for regions similar to annotated regions. The classifier should function in an on-line fashion and must not require long re-training phases. Additionally, the classifier should be able to function with minimal training data. This combination of demands suggests a simple k-nearest-neighbor (kNN) classifier [Duda et al., 2000]. Such a classifier stores all training feature vectors in a matrix. It classifies each region by calculating the Euclidian distance $d$ to all vectors in the training matrix and selecting the $k$ closest training examples which each represent an example of a single class. A simple majority voting system is used to determine the best class for the region. To create a distribution over multiple classes we count the number of occurrences a class in the top $k = 5$ and divide this by $\sum d$ to get a number indicating the match.

System overview

An overview of the annotation system is given in figure 5.1. The system loads, pre-processes, and segments the data of a single file and presents the
Figure 5.2: The cochleogram of several passing cars. Darker means more energy. Solid lines denote manual annotation. Dashed lines denote automatic classification, the dash-dotted lines denote still unclassified regions. The cars are segmented in the pink boxes. A bird is segmented in the green box. The purple boxes are not (yet) annotated.
### Table 5.1: Region feature vector description

<table>
<thead>
<tr>
<th>Feature</th>
<th>Dim</th>
<th>Percentile or range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>1</td>
<td>&gt; 0.02</td>
<td>Fraction of spectro-temporal area equivalent to 1 s</td>
</tr>
<tr>
<td>Channel mean</td>
<td>1</td>
<td>1 - 100</td>
<td>Average channel number (1 is highest, 100 is lowest). This corresponds to average log-frequency contribution.</td>
</tr>
<tr>
<td>Channel std</td>
<td>1</td>
<td>&lt; 50</td>
<td>Provides a single number indication of the channel spread.</td>
</tr>
<tr>
<td>Fast-to-Slow-Ratio</td>
<td>7</td>
<td>[ 5 10 25 50 75 90 95 ]</td>
<td>The distribution of Fast-to-Slow-percentiles provides information about the distribution of strong foreground values</td>
</tr>
<tr>
<td>TF</td>
<td>7</td>
<td>[ 5 10 25 50 75 90 95 ]</td>
<td>The distribution of TF values provides information about the distribution of strong sinusoidal contributions.</td>
</tr>
<tr>
<td>PF</td>
<td>7</td>
<td>[ 5 10 25 50 75 90 95 ]</td>
<td>The distribution of PF values provides information about the distribution of strong pulse-like contributions.</td>
</tr>
<tr>
<td>Channel distribution</td>
<td>7</td>
<td>[ 5 10 25 50 75 90 95 ]</td>
<td>The channel distribution provides more detailed information about the pattern of contributing channels.</td>
</tr>
<tr>
<td>Channel spread</td>
<td>3</td>
<td>5-95, 10-90, 25-75</td>
<td>Provides more detailed information about the channel spread as the difference in channel numbers between three percentile pairs of the channel distribution</td>
</tr>
<tr>
<td>Frame spread</td>
<td>3</td>
<td>5-95, 10-90, 25-75</td>
<td>Provides more detailed information about the temporal spread as the difference in frame numbers between three percentile pairs of the frame distribution</td>
</tr>
</tbody>
</table>
5.3 Results and discussion

Measuring the performance of the system in meaningful numbers is difficult. A sensible measure is the time saved by this system compared to full manual annotation of start and stop times of the sound events. However each annotation session will result in different annotations due to the reasons formulated in the introduction. This makes a fair comparison difficult. Furthermore the current system is not yet sufficiently user-friendly to allow a good comparison. Alternatively we measured how often the correct class was suggested by the kNN classifier. The results are shown in table 5.2. When a class is either not annotated yet or misclassified, it is marked as “alphabetical”, otherwise it is ranked as first or second. Without automated annotations one expects an average rank equal to half the number of classes. Note that with \( k = 5 \) it is possible to have 5 different classes in the list, but third, fourth or fifth ranked classes did not occur in the test. The current system is a first installment of the annotation tool. Its initial performance is encouraging, but each aspect can and must be improved before it is truly useful. The further improvement of the tool will depend strongly on an improved understanding of the annotation process, which in turn is a special form of listening. Initial experience with assisted annotation indicates that result to the user. First, the user selects a region. The selected region can be played as sound and a matching class can either be selected from a class-list or added to the class-list. Initially the list is ordered alphabetically, but when sufficiently matching examples of the class have been encountered, the top-positions on the list will be ordered according to class-likelihood. After class assignment, the kNN training matrix is extended with the feature vector of the region. If the match of a class exceeds a threshold (here set to \( p = 0.04 \)), it is automatically classified as that class. If the match exceeds 0.01, the region will be conditionally classified, which entails that the user has to accept the classification before it is included in the kNN training matrix. Regions that end up without annotation are discarded after the user decides that the file is annotated in sufficient detail. To measure the performance of the system we track the class-rank of manually annotated regions, the number of automatically annotated regions, and the number of accepted regions. The number of discarded regions is a measure for the performance of the segmentation. The final output of the system is a list of classes assigned to regions.

### Table 5.2: Results of an annotation session on the Assen dataset (N = 101)

<table>
<thead>
<tr>
<th></th>
<th>alphabetical</th>
<th>first</th>
<th>second</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>rank</td>
<td>15%</td>
<td>74%</td>
<td>13%</td>
<td>100%</td>
</tr>
</tbody>
</table>

Note that with \( k = 5 \) it is possible to have 5 different classes in the list, but third, fourth or fifth ranked classes did not occur in the test.
the annotator does not analyze the file from start to end, but instead prefers to focus either on individual environmental processes or on individual auditory streams. This allows maximal benefit from process/stream dependent knowledge. It is possible that everyday listening (Gaver, 1993b) reflects this so that at most one stream is analyzed with all available knowledge: the focus of auditory attention. All other streams are analyzed in less detail. This observation in combination with and the annotation problems formulated in this chapter suggest that the question “What do we do when we listen” should become a focus of active research.