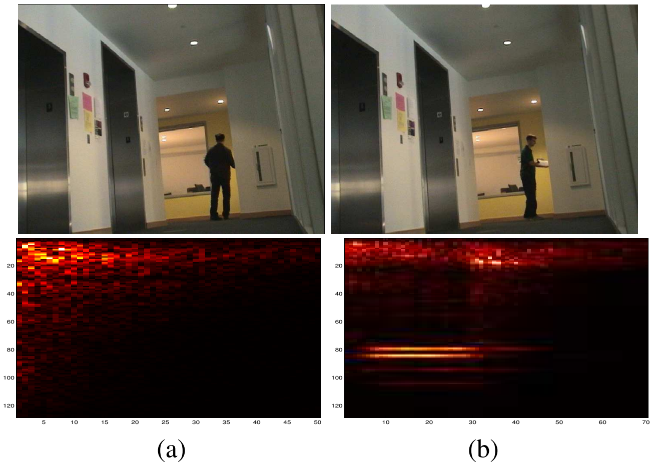
Automated Audio-Visual Activity Analysis

Chris Stauffer
Massachusetts Institute of Technology
Computer Science and Artificial Intelligence Laboratory

Abstract

Current computer vision techniques can effectively monitor gross activities in sparse environments. Unfortunately, visual stimulus is often not sufficient for reliably discriminating between many types of activity. In many cases where the visual information required for a particular task is extremely subtle or non-existent, there is often audio stimulus that is extremely salient for a particular classification or anomaly detection task. Unfortunately unlike visual events, independent sounds are often very ambiguous and not sufficient to define useful events themselves. Without an effective method of learning causally-linked temporal sequences of sound events that are coupled to the visual events, these sound events are generally only useful for independent anomalous sounds detection, e.g., detecting a gunshot or breaking glass. This paper outlines a method for automatically detecting a set of audio events and visual events in a particular environment, for determining statistical anomalies, for automatically clustering these detected events into meaningful clusters, and for learning salient temporal relationships between the audio and visual events. This results in a compact description of the different types of compound audio-visual events in an environment.



(a) (b)

Figure 1: This figure shows a single frame of video extracted from sequences when two separate individuals were tracked as they passed into a secure area. The first individual unlocks the door and passes through resulting in a sound event of the door lock mechanism unlatching shown in the left audio spectrogram. The second individual forces the magnetic door open, resulting in a shrill alarm sound shown in the right spectrogram.

1. Introduction

The field of computer vision has made great strides in making functional activity understanding systems, but these systems are generally deaf to the world around them. Figure 1 shows two individuals trying to gain access to a secure area. One of them unlocks the door, while the other forces the door open. There is almost no information in the visual signal to discriminate these two activities, but the audio contains an anomalous alarm sound, resulting a short time after the unauthorized access. Even trained security personnel would have difficulty differentiating these two events over closed-circuit TV with no audio.

Visual observation allows objects to be tracked from one location to another and allows objects' appearances and activities to be characterized over time. Audio observation compliments visual observation in many ways.

Audio capture is less expensive and less cumbersome than video capture. Audio contains information that is often not available in the visual signal. Audio does not suffer from line-of-sight or occlusion restrictions. This enables an audio system to "see around corners", but can also make it difficult in some cases to separate temporally overlapping audio events. The ability to "see around corners" can enhance visual surveillance applications in the areas that are the most difficult to completely monitor and are often prone to danger, e.g., stairwells, dark corners, tunnels, etc. In many of these situations, the sounds of doors opening, doors closing, footsteps, elevator sounds, unusual sounds, and other auditory cues can be extremely useful in understanding activity and detecting potentially hazardous or risky behavior.

In this paper, we describe a system that is able to detect discrete audio and visual events, determine anomalous audio and visual events, cluster the audio and video events into meaningful classes, and determine the salient temporal chains of these events that correspond to particular activities in the environment. This system can be deployed in any environment with reasonably sparse audio and visual input and is capable of learning complex models of activity involving temporal sequences of particular stimuli.

1.1. Previous Work

In order to automatically learn temporal relationships between classes of visual or audio events, it is first necessary to automatically model the classes of visual and audio events.

Johnson and Hogg [7] clustered activities into 400 clusters based on the object position and direction. Stauffer and Grimson [14] accomplished a similar goal using a hierarchical clustering. This paper exploits two of the most significant events in visual tracking– entrances and exits from an environment. Stauffer presented a automated clustering source and sink locations [12], which is similar to an aspect of this work.

While there has been a large amount of work in the area of speech recognition (see [11]), relatively little work has been focused on general audio understanding. By far the most common general audio application is detecting speech versus non-speech sounds. This is most often accomplished by adaptive thresholds on signal or spectral power levels, i.e., if there is a noise, it must be speech. Much more complex audio analysis has been performed in the area of computational audio scene analysis [4, 2]. Unfortunately, the CASA approaches to audio segmentation tend to run a hundreds to thousands of times slower than real-time. This paper introduces a method for detecting audio events in sparse environments with a robust audio background model which requires little computational overhead.

Research in unsupervised approaches to combined audio-visual understanding has taken many forms. In the area of combined audio-visual segmentation, Clarkson et al. [3] clustered ambulatory audio-visual streams to determine classes corresponding to an individual person's location and the Broadcast News Darpa Challenge [1] evoked numerous papers that segmented audio-visual streams based on speaker changes or story changes. There has also been significant work on detecting scene changes in video. Unfortunately, raw segmentation of audio-visual signals usually does not constitute activity understanding, because audio activities often include multiple audio events at different temporal offsets and multiple sequences may occur in overlapping intervals. There has been other work that exploits synchrony to establish statistical correspondence between localized visual representations and an audio signal [6, 5]. While this work can find audio and visual stimulus that are likely to result from the same underlying cause at the same time, these approaches do not attempt to model sequences of events.

This paper introduces a method for detecting discrete audio and visual events, determining anomalous events, clustering the events into meaningful classes, and determining recurring temporal sequences of particular audio and visual events. This is fundamentally different from any previous approach that simply clusters audio-visual streams or learns correspondence between temporally co-occurring audio and video events. This system is able to learn sequences of audio and video events that correspond to meaningful activity classes. These chains encode temporal information, which is essential to classifying certain activities. This system can represent multiple activities occurring during overlapping temporal windows. Finally, this is all accomplished without supervision and is relatively robust to variation in the algorithms parameters.

Section 2 describes how the independent audio and visual events are detected. Section 3 describes the robust classification of the audio and visual events into meaningful clusters of activity. Section 4 describes a method for determining salient temporal relationships between these event classes and determining which events are causally linked in this model. This automated analsysis is performed on thirty minutes of video. Sections 5 and 6 discuss future work and conclusions.

2. Event Detection

The first step in this analysis is the automated detection of visual and audio events. The majority of the audio and visual events tend to correspond to objects at particular locations performing particular actions. The following two subsections describe the visual and audio preprocessing required to detect discrete audio and visual events. Both systems are less effective in heavily populated environments with continuously varying ambient audio signals. Fortunately, many visual surveillance applications involve sparsely populated indoor and outdoor settings. The authors anticipate that as our understanding of these problems and the computational prowess with which we face them increase, these limitations will be circumvented with more effective, real-time detection and segmentation of objects and audio events. Section 3 will then describe how these events are clustered into meaningful classes.

2.1. Video Event Detection

Our work uses an implementation of the Adaptive Background Mixture Model of Stauffer and Grimson as described in [13] to track moving objects in sparsely populated environments. Figure 2 shows one hour of tracking data superimposed on the corresponding scene of an elevator lobby.

A quick description of tracking systems based on background subtraction is: characterize the appearance of the

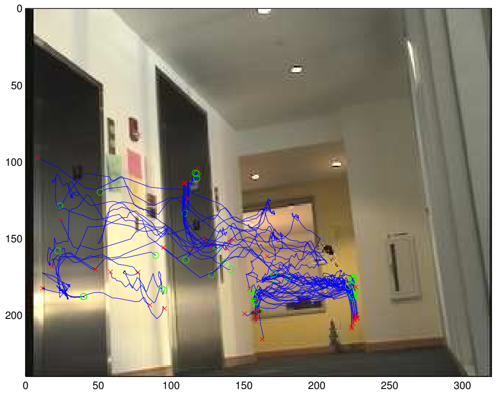


Figure 2: This figure shows con-trails for the tracking data of pedestrians in an elevator lobby. The tracks are blue and begin at locations with green circles and end at locations with red x's. There are three elevators in the left foreground. In the back, bottom of the image, one can see the hallway where many pedestrians pass from left to right or right to left. Though not visible in this camera view, there is a security door at the rightmost location where pedestrians exit.

static (and sometimes repetitive) elements of a visual scene; detect pixels that are not characteristic of the static elements; group those pixels; track the groups of pixels from frame to frame using a set of linear Kalman filters. This approach is one of a few approaches that results in real-time tracking of unrestricted object types in unstructured environments.

These tracks are filtered to remove transient tracks and to smooth trajectories. Finally, the state of the initial and final observations of the tracking sequences, e_i^α, and e_i^ω, are extracted as in [12]. These two types of events correspond to particular activities. It may be possible to extract events that are internal to the scene by finding common locations or locations where loitering occurs, but only the entrance and exit events are used in this work.

2.2. Audio Event Detection

Our method for detecting audio events is somewhat analogous to our adaptive background segmentation. Rather than attempting to characterize the "pixel process", this work attempts to characterize the "audio process". As stated earlier, our approach is appropriate for sparse audio environments, i.e., where "audio events" are generally non-overlapping in time.

Given a sampled audio sequence $x_s(t)$, the Fast-Fourier Transform (FFT) coefficients are calculated on windows of W samples enveloped by a corresponding hamming window function. Computing these FFT coefficients at intervals of $W/2$ results in a new time sequence of FFT coefficients $x_f(t)$. Whereas $x_s(t)$ is a scalar function over time at the sample interval, $x_f(t)$ is a multinomial function over time sampled at a slower rate.

We model the audio process using a mixture of k spectral exemplars, $(S_1, S_2, ..., S_k)$. Each spectral exemplar S_i is parameterized by a mean spectral value μ_i, a variance Σ_i, and a weight w_i. Each incoming sample spectrum is matched to the exemplar that is within k_α standard deviations. This hard matching is based on the Mahalanobis distance in $log(x_f(t))$-space.

If multiple exemplar's match, the exemplar S_i with the highest ratio of weight to variance $(w_i/|\Sigma_i|)$ is chosen as a discrete match. It is possible to use a soft assignment function, but in our experience it results in exemplar drift causing exemplars that are temporarily without supporting evidence to drift towards other exemplars. This is usually undesirable. If no exemplar matches the current sample, the noisiest exemplar with the least recent evidence replaces the a new exemplar at the current sample position with a small weight and large variance. Thus, *every* sample spectrum matches a single exemplar. The weight, mean, and variance of the matching exemplar is updated using an online scheme described in [13]. Our approach differs in that we use different learning rates for the mean, variance, and weight parameters. This allows the mean value to be tracked at a reasonable rate, but the variance and weight can estimate a long-term average. Also, the adaptation rate is constant rather than related to the probability of the match, which increases the stability of this algorithm.

Finally, the adaptation can be scheduled such that the learning rate is decreased as more samples are received. I.e., Initial adaptation rate for mean is 1.0, and it decreases as more samples are received. In our implementation the adaptation rate asymptotes at to a constant factor, which is related to the factor in previous descriptions of this work. The scheduling is different for each adaptation rate (mean, variance, and weight).

The goal of the exemplars is to characterize the entire audio process. This includes the "background process" and the "foreground process". In this case, the background process is what is commonly referred to as background noise, e.g., air conditioners, computer fans, street noise, repetitive beeps, etc. The foreground process refers to non-repetitive transients in the environment. Given the evolving set of exemplars, our system uses the method described in [13] to classify the process models as background or foreground. Each sample $x_f(t)$ is classified as foreground or background based on the exemplar to which it is assigned. The final segmentation results from a median filtered estimate of the foreground binary stream.

In our experience, this method is able to robustly represent relatively complex background noise and quickly adapt to changes in the environmental audio, while not generally corrupting the background model when new sound events occur. Using this method on thirty minutes of video from the scene shown in Figure 2, 262 discrete audio events were detected. These clips can be played back in less than one tenth of the original time by simply removing the background segments. Removing dead space in the signal allows very quick review of the sounds in a particular environment, but often hinders understanding of how discrete sounds combine to form meaningful sequences. Section 3 describes how to derive a compact description of the visual and audio events.

3. Event classification and Anomaly Detection

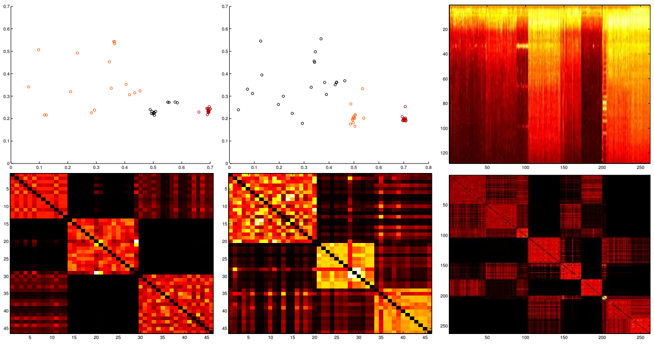
To automatically find meaningful sequences of events, the events must be automatically clustered into homogeneous classes of events. Without the capability to classify the type of an event, it would be impossible to learn meaningful relationships between events. E.g., after someone unlocks the security door, they will *either* appear 5.8 ± 1.4 seconds later from around the corner at the end of the hallway or you will hear the storage door open 2.8 ± 0.4 seconds later. If someone doesn't does not appear when they are expected or if the unobserved door isn't heard re-latching, it may be a security risk. A similar description consisting of terms like after one generic sound another generic sound will occur, is much less descriptive or useful. This section describes an automated method for inferring event classes such as the sound of a door being unlocked or an object leaving the scene at a particular location. This work uses the same clustering algorithm to cluster both the audio descriptors and the visual descriptors.

As described in the previous section, the visual events correspond to the starting and ending states of individual tracking sequences. The source and sink descriptors, d^{α} and d^{ω}, are $[x, y, s]$, where x and y are the normalized image position and s is the square root of the objects projected area relative to the entire image. Significant source and sink clusters, c_i^{α} and c_i^{ω} will generally correspond to the position and size of objects at particular entry and exit locations, e.g., doors, hallways, and permanant visual occlusions.

The audio events correspond to discrete transient sounds in the environment. The audio descriptors d^{a} used in this work are the log of the average magnitude of the Fast-Fourier Transform of enveloped 50ms windowsw within an audio sequence. Significant audio clusters c_i^{a} will generally correspond to sounds from a particular source.

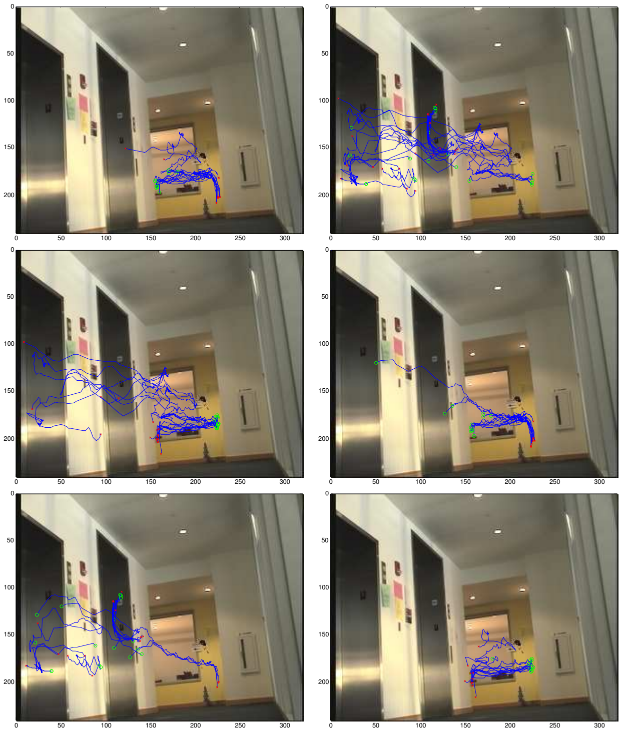
3.1. Event Classification

Given a set of events $\{e_1, e_2, ..., e_n\}$ and descriptions of those events, $\{d_1, d_2, ..., d_n\}$, our goal is to find a set of K informative clusters. The causal link analysis described in the next section benefits from reliable estimation of a set



(a) (b) (c)

Figure 3: This figure shows the source location, sink locations, and average fft features used to cluster the individual events. The second row contains the $s(i,j)$. The rows and columns have been ordered to illustrate the effectiveness of the final clustering for values $K^{\alpha} = 3$, $K^{\omega} = 3$, and $K^{a} = 9$.



(a) (b)

Figure 4: This figure shows the tracking sequences automatically clustered by source location in the left column (a) and sink location in the right column (b).

of clusters that are both homogeneous (containing mostly the same type of event) and also non-redundant (not similar to other clusters). The selection of the number of clusters K can trade-off these two factors, but without robust clustering, no value of K will result in meaningful clusters of events.

We use the spectral clustering algorithm as described in [9] to cluster in both modalities. The similarity of two points is defined as

$$s(d_i, d_j) = \phi_i(d_j), \quad (1)$$

where ϕ_i is a Gaussian with mean $\mu_i = d_i$ and variance σ_i. The variance of the kernel σ_i is calculated as the mean of $\frac{1}{K}^{th}$ lowest percentage of pairwise distances from d_i. Using this variable kernel increased the robustness of the clustering despite widely varying values of K and dimensionalities of input. Spectral clustering with an appropriate kernel can result in effective clustering despite extreme within-class variation when the classes are reasonably separable and the classes are densely connected.

Figure 3 shows the features and similarity matrices for the sources, sinks, and audio events. Figure 4 shows the tracking sequences clustered into three classes automatically by both source state d^{α} and sink state d^{ω}. In this scene, the sources correspond to entering the scene near the elevators to the left, from the hallway in the lower left, and from the security door in the lower right. Source clustering was 93.5% effective and sink state clustering was 95.7% effective.

The original thirty minute audio sequence was relatively sparse, containing only a few hundred discrete audio events. The result of the foreground/background segmentation discussed in the previous section is a set of 262 independent audio events. The audio events corresponded to footfalls, doors opening, doors closing, security lock sounds, security alarm sounds, sounds produced by the elevators, muffled speech[1], and other transient sounds. The audio clips averaged approximately one to two seconds in duration.

The occurrence of an audio event is not extremely informative of the scene activity except that it may indicate that one or more objects may be present in the environment. Thus, without grouping these events into meaningful clusters, very little can be inferred from the audio signal.

Using the spectral clustering technique described above on the audio descriptors d_i^a, we clustered the audio events into nine separate categories, $[c_1^a, c_2^a, ..., c_9^a]$. Figure 5(a)-(i) show the audio events that were clustered into each of the nine clusters.

Table 3.1 contains the confusion matrix for the unsupervised clustering to human-labeled audio event clusters. Most of the errors were due to combined audio events for which the human was forced to choose a class. Because of the number of footsteps and their similarity to other transients, more than one cluster was used to represent audio events corresponding to footsteps, bumps, and similar transients. Extremely distinct classes such as the security door alarm were clustered very effectively. In order for humans to effectively differentiate between the audio clusters for doors opening, doors closing, and doors re-latching, it was necessary for to watch the video tape. In that light, the clustering performance was much better than expected.

3.2. Anomaly Detection

The likelihood of each source or sink event under a Parzen density estimator is sufficient to characterize the likelihood of a particular event.

$$L(e_i) = \sum_{j \neq i} w_j \phi_j(e_i), \quad (2)$$

where w_i is a normalized weighting of the events and ϕ_j is the same function used to define our similarity metric. Anomalies are statistical outliers. In the case of sources and sink events, there are no significant anomalies.

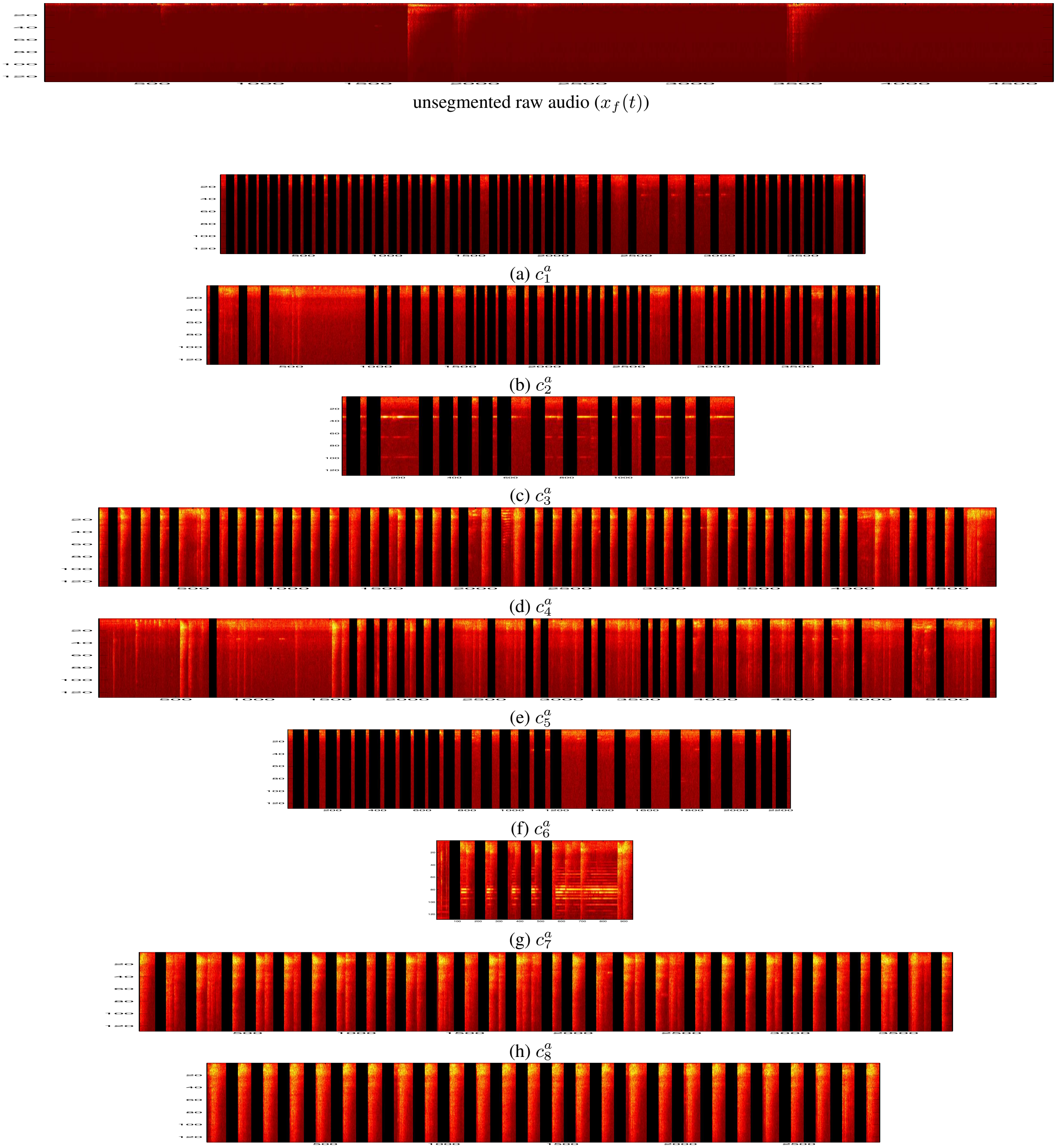
4. Causal Link Analysis

This section describes how to determine causal links between different events in a temporal stream. At this point it is no longer necessary to differentiate between audio and visual events, only unique event classes. Each event is represented only by its class type and the time it occurred. In our previous example, there are 15 unique classes ($K = K^{\alpha} + K^{\omega} + K^{a}$).

Given a set of N events $\{e_1, e_2, ..., e_N\}$, the classes of those events $\{c_1, c_2, ..., c_N\}$, and the onset times of the events $\{t_1, t_2, ..., t_N\}$, it is possible to determine salient temporal relationships between event pairs and determine chains of events that correspond to regular activities in a particular environment. This is done without supervision.

To limit the complexity of the problem, we have chosen to represent only temporal chains of events, i.e., sequences of events with no forks. Each event in a chain can at most one element before directly proceeding it and at most one element directly coupled after it in the chain. Thus, if an event initiates a set of events, those events must be represented in a single chain of events. In most cases, a chain can effectively represent activities, because they often occur in the same temporal order. For example, if someone entering a store causes a chime *and* the door to close, both events usually occur with the same relative timing. Thus, a chain of "entrance" → "chime" → "door closing" is a reasonable approximation.

[1] The audio was physically filtered to avoid the possibility that conversations could be understood.



unsegmented raw audio ($x_f(t)$)

(a) c_1^a

(b) c_2^a

(c) c_3^a

(d) c_4^a

(e) c_5^a

(f) c_6^a

(g) c_7^a

(h) c_8^a

(i) c_9^a

Figure 5: This figure shows unsegmented raw audio containing four separate events (top) and the concatenation of all of the audio events from each of nine audio clusters (a)-(h). The clusters roughly correspond to footsteps and transient conversations and noises (a-b), elevator pings (c), the security door re-latching (d-e), security door alarms (f), security alarm sounding (g), security door closing (h), and security door being opened (i).

Description	c_1^a	c_2^a	c_3^a	c_4^a	c_5^a	c_6^a	c_7^a	c_8^a	c_9^a
security door opening (SDO)	0	0	0	1	2	0	0	29	0
security door closing (SDC)	0	0	0	2	0	0	0	0	26
security door relatching (SDR)	0	0	0	15	0	0	0	2	0
security door alarm (SDA)	0	0	0	0	0	0	5	0	0
elevator pings (EP)	3	0	15	1	0	2	0	0	0
elevator opening (EO)	0	5	0	2	19	0	0	0	0
footsteps (F)	44	18	0	0	1	24	0	0	0
other transients (T)	0	3	0	1	5	1	1	0	0

Table 1: This table shows the confusion matrix for the automated clustering of audio events.

Automatically finding temporal relationships and determining events that are included in a temporal chain involves two steps: determining the most likely correspondence chains; and estimating the linking likelihoods. Unfortunately, neither of these steps can be solved independently of the other.

4.1. Inferring the Correspondence Chains

Suppose that an oracle provided reasonable estimates of the likelihood of a pair of events (parameterized event classes and relative times), given that the two events were linked

$$p(c_i, c_j, \delta t_{ij} | \gamma_{ij} = 1) \tag{3}$$

where $\hat{\gamma}_{ij}$ is an indicator variable that is zero if the occurrence of the first event is not directly responsible for the occurrence of the second event. The oracle also provided an estimate of the likelihood of a pair of events given the event classes and times given that the two events were *not* linked

$$p(c_i, c_j, \delta t_{ij} | \gamma_{ij} \neq 1). \tag{4}$$

These two values together with an estimate of the relative likelihood of linked versus not linked pairs is sufficient to estimate the posterior

$$p(\gamma_{ij} = 1 | c_i, c_j, \delta t_{ij}) \tag{5}$$

using Bayes Rule.

Given the estimate of the posterior likelihood of $gamma_{ij} = 1$ for all i, j pairs of events, optimal chaining hypothesis Γ^* is defined as

$$\Gamma^* = \underset{\Gamma}{argmax} \left[L(\Gamma) = \prod_{i,j:\gamma_{ij}=1} p(\gamma_{ij} = 1 | c_i, c_j, \delta t_{ij}) \right]. \tag{6}$$

where all valid Γ hypotheses must obey the chaining restriction described above.

Rather than using a Markov Chain Monte Carlo (MCMC) approximation as was previously done in [10], we solve for the optimal solution using a variant of the Hungarian Algorithm [8]. Our variant allows for each event to be result from a *null* event and be terminated by a *null* event. The likelihood of these *null* events, $p(\gamma_{\alpha j} | c_\alpha, c_j)$ and $p(\gamma_{i\omega} | c_i, c_\omega)$, are assumed to be a constant and is a parameter of this system.

Unfortunately, our automated system is not provided with an estimate of linking likelihoods, so the must be estimated.

4.2. Estimating the Linking Likelihoods

Suppose an oracle provided a reasonable linking hypothesis $\hat{\Gamma}$. Given this linking hypothesis, we estimate the likelihood of the linking potentials by estimating $p(c_i, c_j, \delta t_{ij} | \gamma_{ij} = 1)$ using the Parzen density

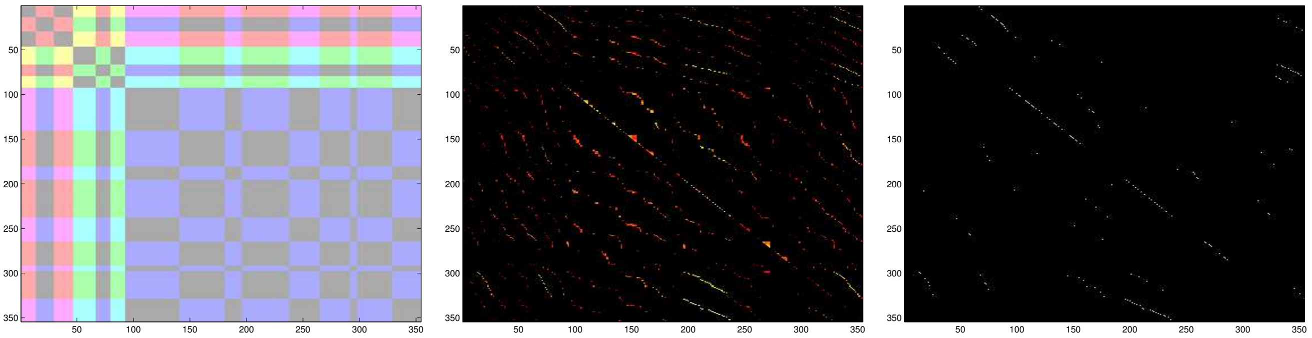
$$p(c_i, c_j, \delta t_{ij} | \gamma_{ij} = 1) = \sum_{i,j} \frac{w_{ij}}{\sum_{i,j} w_{ij}} \psi_{c_i, c_j}(\delta t_{ij}), \tag{7}$$

where w_{ij} is one if γ_{ij} is one and zero otherwise. $\psi_{c_i, c_j}(\delta t_{ij})$ is a Gaussian over the inter-arrival time with variance of one second. A variance of about one second is reasonable for most surveillance applications. $p(c_i, c_j, \delta t_{ij} | \gamma_{ij} \neq 1)$ is calculated in the same way.

Unfortunately, our automated system is not provided with an initial linking hypothesis.

4.3. Iterative Optimization

Our automated system begins with no assignment and no linking likelihood estimates. Our initial estimate of the linking likelihoods (Equation 7) is estimated as an expectation over all possible assignments where assignments have not been determined. This is equivalent to Equation 7 except that the values of w_{ij} are a normalized weighting of the pairs of samples. Thus, our first estimation of Equation 7 assumes that each event pair is equally likely to be assigned to any other event. As the more assignments are made, the linking likelihood estimate becomes a better approximation of the ideal linking likelihood estimate.



(a) (b) (c)

Figure 6: This figure shows three $356x356$ matrices. In the first matrix (a), the red rows and columns represent sets of event pairs either beginning with a source event or ending with a source event respectively. The green rows and columns represent the three classes of sink events and the blue rows and columns represent the nine audio event clusters. The second matrix (b) represents the estimate of $p(c_i, c_j, \delta t_{ij} | \gamma_{ij} = 1)$. The final matrix represents the optimal assignment for (b) as defined in Equation 6.

After this initial estimate is made the iterative procedure continues as follows:

- Estimate the optimal linking hypothesis $\hat{\Gamma}$, given the current linking likelihood estimates.
- Estimate the linking likelihoods given the current linking hypothesis $\hat{\Gamma}$.

4.4. Linking Results

Figure 6 shows the three matrices for the first iteration of assignment. The first matrix shows which rows and columns corresponding to the different audio and video event classes. The second matrix shows the assignment likelihoods as estimated in the first iteration. The final matrix shows the optimal assignment for this iteration. After the first iteration, the most salient causal relationships are strengthened in the link likelihood estimation. As the iterative estimation continues the causal relationships become more and more salient. After 3-5 iterations, the process tends to stabilize, although reasonable results are achieved after the second iteration.

From the 356 individual events over 109 causal chains were inferred. There were 20 chains that contained minor variations of:

- enter from right,
- exit to left 3.4 seconds later,
- door closes 4.2 seconds later,
- first door re-latches 19.8 seconds later,
- second door re-latches 2.02 seconds later.

Other significant chains included: series of footsteps separated by a second or two in initial offset time; regular visual entrance and visual exit pairs[2]; exiting through the security doors and the doors subsequently closing and latching; the elevator opening followed by a series of footsteps and sometimes the security door being opened; etc. Of the chains that were found, the average chain length was 3.3 events. The longest chain was 8 events in length.

This chaining results in a compact description of events that are otherwise disassociated. In the example shown, our system learned when objects are expected to appear based purely on audio evidence as well as what one might expect to hear after objects have left the field-of-view. Objects whose track was lost may be able to be stitched together if their entrance and exits are detected reliably and exhibit low entropy. Finally, this system is capable of working when visual evidence is not available due to lost cameras or lack of cameras to begin with.

5. Future Work

The most obvious area for future investigation is to integrate clustering into the iterative optimization procedure. There are many events that have similar likelihoods under two different classes. Using the temporal context, it may be possible to cluster the events more effectively. This will result in more salient temporal chaining and better activity models.

The most significant parameter in this work is the number of clusters for each modality. Automatic model selection is a difficult problem as has been pointed out by many researchers. It is our hope that it will be possible to robustly estimate the number of clusters automatically, given effective clustering *and* a model of temporal relationships.

Finally, we intend to investigate the stability of this algorithm in extended scenes with multiple sensors. This type of approach has shown promise in learning linkage across non-overlapping visual sensors. With the addition of intervening audio events, we

[2] It is worth noting that these event pairs would be detected even when the tracking was lost as long as the source event and sink event were predictive of one another.

believe this algorithm may prove very valuable for tracking objects through extended environments.

6. Conclusions

This paper introduced a novel activity analysis paradigm. We have shown that our system is capable of detecting discrete audio and visual events, determining anomalous events, clustering the events into meaningful classes, and determining recurring temporal sequences of particular audio and visual events. This approach fundamentally differs from any previous approach that simply clusters audio-visual streams or learns correspondence between temporally co-occurring audio and video events.

This system is able to learn sequences of audio and video events that correspond to meaningful activity classes. These chains encode temporal information, which is essential to classifying the activities. This system can represent multiple activities occurring during overlapping temporal windows. Finally, this is all accomplished without supervision and is relatively robust to variation in the algorithms parameters.

References

[1] *Proceedings of DARPA Broadcast News Transcription Understanding Workshop*, Lansdowne, Virginia, February 8-11 1998.

[2] Albert S. Bregman, editor. *Auditory Scene Analysis: the perceptual organization of sound*. MIT Press, Cambridge, MA, 1990.

[3] Brian P Clarkson and Alex Pentland. Unsupervised clustering of ambulatory audio and video. In *Proceedings of IEEE International Conference On Acoustics, Speech, and Signal Processing*, volume 6, pages 3037–3040, Phoenix, Arizona, March 15-19 1999.

[4] D.P.W. Ellis. Prediction-driven computational auditory scene analysis for dense sound mixtures. In *Prediction-driven computational auditory scene analysis for dense sound mixtures*, Keele, July 1996.

[5] John Hershey and Javier R. Movellan. Audio vision: Using audio-visual synchrony to locate sounds. In *Proceedings of the 13th Annual Conference in Neural Information Processing Systems (NIPS)*, 1999.

[6] John W. Fisher III, Trevor Darrell, William T. Freeman, and Paul Viola. Learning joint statistical models for audio-visual fusion and segregation. In Tom Dietterich and MIT Press (2001) Volker Tresp, editors, *Proceedings of the 13th Annual Conference in Neural Information Processing Systems*, 2000.

[7] Neil Johnson and David C. Hogg. Learning the distribution of object trajectories for event recognition. In Pycock D., editor, *British Machine Vision Conference (BMVA)*, pages 583–592, September 1995.

[8] H. W. Kuhn. The hungarian method for solving the assignment problem. In *Naval Research Logistics Quarterly*, volume 2, pages 83–97, 1955.

[9] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, pages 849–856, MIT Press. Cambridge, MA, 2002.

[10] Hanna Pasula, Stuart J. Russell, Michael Ostland, and Yaacov Ritov. Tracking many objects with many sensors. In *IJCAI99*, pages 1160–1171, 1999.

[11] Lawrence Rabiner and Biing-Hwang Juang. *Fundamentals of Speech Recognition*. Prentice Hall, Inc., New Jersey, 1993.

[12] Chris Stauffer. Estimating tracking sources and sinks. In *Proc. Event Mining Workshop*, Madison, WI, July 2003. IEEE Press.

[13] Chris Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition*, pages 246–252, 1999.

[14] Chris Stauffer and W. E. L. Grimson. Automatic hierarchical classification using time-based co-occurrences. In *Computer Vision and Pattern Recognition*, pages 333–339, 1999.