Visually Indicated Sounds

Andrew Owens[1]
Phillip Isola[2,1]
Josh McDermott[1]
Antonio Torralba[1]
Edward H. Adelson[1]
William T. Freeman[1]

[1]Massachusetts Institute of Technology
[2]University of California, Berkeley

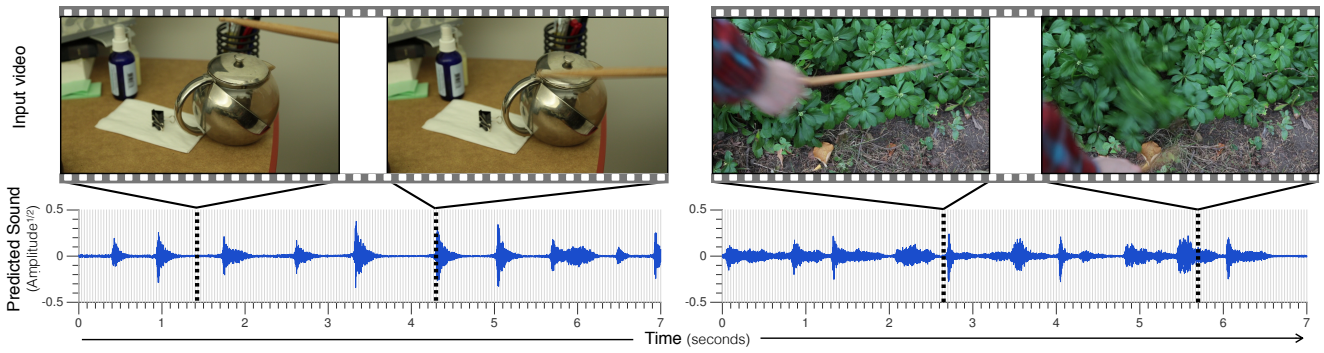


Figure 1: We train a model to synthesize plausible impact sounds from silent videos, a task that requires implicit knowledge of material properties and physical interactions. In each video, someone probes the scene with a drumstick, hitting and scratching different objects. We show frames from two videos and below them the predicted audio tracks. The locations of these sampled frames are indicated by the dotted lines on the audio track. The predicted audio tracks show seven seconds of sound, corresponding to multiple hits in the videos.

Abstract

Materials make distinctive sounds when they are hit or scratched – dirt makes a thud; ceramic makes a clink. These sounds reveal aspects of an object's material properties, as well as the force and motion of the physical interaction. In this paper, we introduce an algorithm that learns to synthesize sound from videos of people hitting objects with a drumstick. The algorithm uses a recurrent neural network to predict sound features from videos and then produces a waveform from these features with an example-based synthesis procedure. We demonstrate that the sounds generated by our model are realistic enough to fool participants in a "real or fake" psychophysical experiment, and that they convey significant information about the material properties in a scene.

1. Introduction

From the clink of a porcelain mug placed onto a saucer, to the squish of a shoe pressed into mud, our days are filled with visual experiences accompanied by predictable sounds. On many occasions, these sounds are not just statistically associated with the content of the images – the way, for example, that the sounds of unseen seagulls are associated with a view of a beach – but instead are directly caused by the physical interaction being depicted: you *see* what is making the sound.

We call these events *visually indicated sounds*, and we propose the task of predicting sound from videos as a way to study physical interactions within a visual scene (Figure 1). To accurately predict a video's held-out soundtrack, an algorithm has to know something about the material properties of what it is seeing and the action that is being performed. This is a material recognition task, but unlike traditional work on this problem [3, 34], we never explicitly tell the algorithm about materials. Instead, it learns about them by identifying statistical regularities in the raw audiovisual signal.

We take inspiration from the way infants explore the physical properties of a scene by poking and prodding at the objects in front of them [32, 2], a process that may help them learn an intuitive theory of physics [2]. Recent work suggests that the sounds objects make in response to these interactions may play a role in this process [35, 38].

We introduce a dataset that mimics this exploration process, containing hundreds of videos of people hitting, scraping, and prodding objects with a drumstick. To synthesize sound from these videos, we present an algorithm that uses

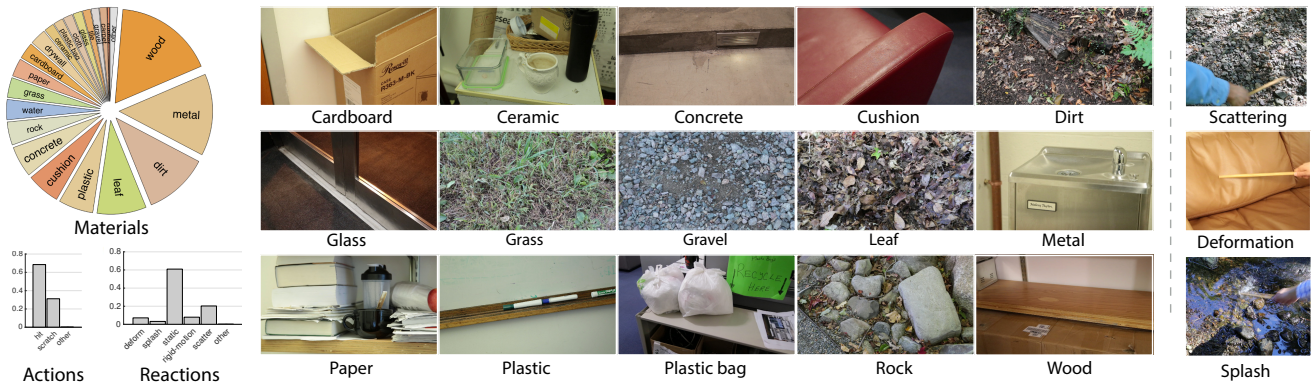


Figure 2: *Greatest Hits Volume 1* dataset. What do these materials sound like when they are struck? We collected 978 videos in which people explore a scene by hitting and scratching materials with a drumstick, comprising 46,620 total actions. We labeled the actions with material category labels, the location of impact, an action type label (hit versus scratch), and a reaction label (shown on right). These labels were used only in analysis of what our sound prediction model learned, not for training it. We show images from a selection of videos from our dataset for a subset of the material categories (here we show examples where it is easy to see the material in question).

a recurrent neural network to map videos to audio features. It then converts these audio features to a waveform, either by matching them to exemplars in a database and transferring their corresponding sounds, or by parametrically inverting the features. We evaluate the quality of our predicted sounds using a psychophysical study, and we also analyze what our method learned about actions and materials through the task of learning to predict sound.

2. Related work

Our work closely relates to research in sound and material perception, and to representation learning.

Foley The idea of adding sound effects to silent movies goes back at least to the 1920s, when Jack Foley and collaborators discovered that they could create convincing sound effects by crumpling paper, snapping lettuce, and shaking cellophane in their studio[1], a method now known as Foley. Our algorithm performs a kind of automatic Foley, synthesizing plausible sound effects without a human in the loop.

Sound and materials In the classic mathematical work of [23], Kac showed that the shape of a drum could be partially recovered from the sound it makes. Material properties, such as stiffness and density [33, 27, 13], can likewise be determined from impact sounds. Recent work has used these principles to estimate material properties by measuring tiny vibrations in rods and cloth [6], and similar methods have been used to recover sound from high-speed video of a vibrating membrane [7]. Rather than using a camera as an instrument for *measuring* vibrations, we infer a plausible sound for an action by recognizing what kind of sound this action would normally make in the visually observed scene.

[1]To our delight, Foley artists really do knock two coconuts together to fake the sound of horses galloping [4].

Sound synthesis Our technical approach resembles speech synthesis methods [26] that use neural networks to predict sound features from pre-tokenized text features and then generate a waveform from those features. There are also methods for generating impact sounds from physical simulations [40], and with learned sound representations [5]. However, it is unclear how to apply these methods to our problem setting, since we train on unlabeled videos.

Learning visual representations from natural signals Previous work has explored the idea of learning visual representations by predicting one aspect of the raw sensory signal from another. For example, [9] learned image features by predicting the spatial relationship between image patches, and [1, 20] by predicting the relative camera pose between frames in a video. Several methods have also used temporal proximity as the supervisory signal [29, 16, 42, 41]. Unlike these approaches, we learn to predict one sensory modality (sound) from another (vision). There has also been other work that trained neural networks from multiple modalities. For example, [30] learned a joint model of sound and vision. However, while they study speech using an autoencoder, we focus on material interaction and use a recurrent neural network to regress sound from video.

A central goal of other methods has been to use a proxy signal (*e.g.* temporal proximity) to learn a generically useful representation of the world. In our case, we predict a signal – sound – known to be a useful representation for many tasks [13, 33], and we show that the output (*i.e.* the predicted sound itself, rather than some internal representation in the model) is predictive of material and action classes.

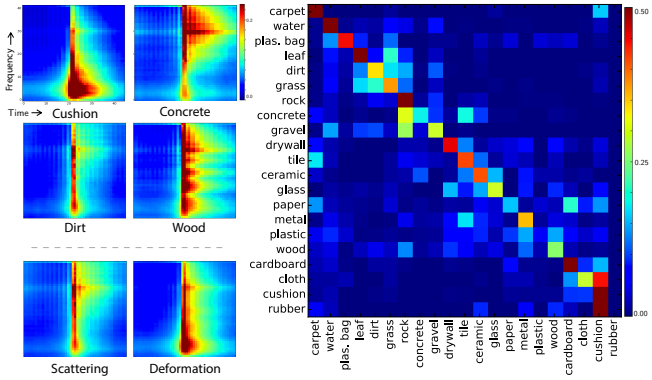
3. The *Greatest Hits* dataset

In order to study visually indicated sounds, we collected a dataset of videos of a human probing environments with a drumstick – hitting, scratching, and poking different objects in the scene (Figure 2). We chose to use a drumstick so that we could have a consistent way of generating the sounds. A drumstick is also narrow and thus does not occlude much of the scene, which makes it easier to see what happens after the impact. This motion, which we call a *reaction*, can be important for inferring material properties – a soft cushion will deform significantly more than a firm cushion, and the sound will correspondingly be different as well. Similarly, individual pieces of gravel and leaves will scatter when they are hit, and their sound will vary according to this motion (Figure 2, right).

Unlike traditional object- or scene-centric datasets, such as ImageNet [8] or Places [43], where the focus of the image is a full scene, ours contains close-up views of a small number of objects. These images reflect the viewpoint of an observer who is focused on the interaction taking place; they contain enough detail to see fine-grained texture and the reaction that occurs after the interaction. In some cases, only part of an object is visible, and neither its identity nor other high-level aspects of the scene are easily discernible. Our dataset is also similar to work in robotics [31, 14] where a robot manipulates objects in its environment. By having a human collect the data instead, we can quickly capture a large number of interactions in real-world scenes.

We captured 978 videos from indoor (64%) and outdoor scenes (36%). The outdoor scenes often contain materials that scatter and deform, such as grass and leaves, while the indoor scenes contain a variety of hard materials, such as metal and wood. Each video, on average, contains 48 actions (approximately 69% hits and 31% scratches) and lasts 35 seconds. We recorded sound using a shotgun microphone attached to the top of the camera, with a wind cover for outdoor scenes. To increase the quality of the recordings, we used a separate audio recorder without auto-gain, and we applied a denoising algorithm [18] to each audio track.

We also collected semantic annotations for a sample of impacts using online workers from Amazon Mechanical Turk (63% of impacts were labeled this way). These included material labels, action labels (hit *vs.* scratch), reaction labels, and the pixel location of each impact site. The distribution of these labels (per impact) is shown in Figure 2. We emphasize that the annotations were used only for analysis: our algorithm was trained from raw videos. Examples of several material and action classes are shown in Figure 2. We include more details about our dataset in Appendix A3.



(a) Mean cochleagrams (b) Sound confusion matrix

Figure 3: (a) Cochleagrams for selected categories. We extracted audio centered on each impact sound in the dataset and computed our subband-envelope representation (Section 4), then computed the average for each category. The differences between materials and reactions are visible: *e.g.*, cushion sounds tend to carry a large amount of energy in low-frequency bands. (b) Confusion matrix derived from classifying sound features. The ordering was determined by clustering the rows of the confusion matrix, which correspond to the confusions made for each ground-truth class.

4. Sound representation

Following work in sound synthesis [28, 37], we get our sound features by decomposing the waveform into subband envelopes – a simple representation obtained by filtering the waveform and applying a nonlinearity. We apply a bank of 40 band-pass filters spaced on an equivalent rectangular bandwidth (ERB) scale [15] (plus a low- and high- pass filter) and take the Hilbert envelope of the responses. We then downsample these envelopes to 90Hz (approximately 3 samples per frame) and compress them. More specifically, we compute envelope $s_n(t)$ from a waveform $w(t)$ and a filter f_n by taking:

$$s_n = D(|(w * f_n) + jH(w * f_n)|)^c, \qquad (1)$$

where H is the Hilbert transform, D denotes downsampling, and the constant $c = 0.3$.

The resulting representation is known as a *cochleagram*. In Figure 3(a), we visualize the mean cochleagram for a selection of material and action categories. This reveals, for example, that cushion sounds tend to have more low-frequency energy than those of concrete.

How well do impact sounds capture material properties in general? To measure this empirically, we trained a linear SVM to predict material category ground-truth sounds in our database, using the subband envelopes as our feature vectors. Before training, we resampled the dataset so that each category had no more than 300 examples. The resulting material classifier has 40.0% balanced class accuracy, and the confusion matrix is shown in Figure 3(b). At the

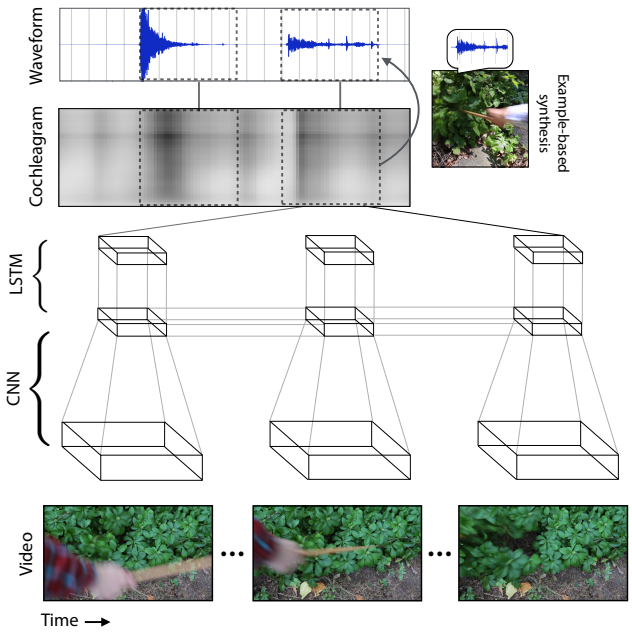


Figure 4: We train a neural network to map video sequences to sound features. These sound features are subsequently converted into a waveform using parametric or example-based synthesis. We represent the images using a convolutional network, and the time series using a recurrent neural network. We show a subsequence of images corresponding to one impact.

same time, there is a high degree of confusion between materials that make similar sounds, such as cushion, cloth, and cardboard, and also concrete and tile.

These results suggest that sound conveys significant information about material, and that if an algorithm could learn to accurately predict sounds from video, then it would have implicit knowledge of these properties. We now describe how to infer these sound features from video.

5. Predicting visually indicated sounds

We formulate our task as a regression problem – one where the goal is to map a sequence of video frames to a sequence of audio features. We solve this problem using a recurrent neural network that takes color and motion information as input and predicts the the subband envelopes of an audio waveform. Finally, we generate a waveform from these sound features. Our neural network and synthesis procedure are shown in Figure 4.

5.1. Regressing sound features

Given a sequence of input images $I_1, I_2, ..., I_N$, we would like to estimate a corresponding sequence of sound features $\vec{s}_1, \vec{s}_2, ..., \vec{s}_T$, where $\vec{s}_t \in \mathbb{R}^{42}$. These sound features correspond to the cochleagram shown in Figure 4. We solve this regression problem using a recurrent neural network (RNN) that takes image features computed with a convolutional neural network (CNN) as input.

Image representation We found it helpful to represent motion information explicitly in our model using a two-stream approach [10, 36]. While two-stream models often use optical flow, we found it difficult to obtain accurate flow estimates due to the presence of fast, non-rigid motion. Instead, we compute *spacetime* images for each frame – images whose three channels are grayscale versions of the previous, current, and next frames. Derivatives across channels in this model correspond to temporal derivatives, similar to 3D video CNNs [24, 21].

For each frame t, we construct an input feature vector x_t by concatenating CNN features for both the spacetime image and first color image[2]:

$$x_t = [\phi(F_t), \phi(I_1)], \tag{2}$$

where ϕ are CNN features obtained from layer fc_7 of the AlexNet architecture [25], and F_t is the spacetime image at time t. In our experiments (Section 6), we either initialized the CNN from scratch and trained it jointly with the RNN, or we initialized with weights from a network trained for ImageNet classification. When we used pretraining, we precomputed the features from the convolutional layers for speed and fine-tuned only the fully connected layers.

Sound prediction model We use a recurrent neural network (RNN) with long short-term memory units (LSTM) [17] that takes CNN features as input. To compensate for the difference between the video and audio sampling rates, we replicate each CNN feature vector k times, where $k = \lfloor T/N \rfloor$ (we use $k = 3$). This results in a sequence of CNN features $x_1, x_2, ..., x_T$ that is the same length as the sequence of audio features. At each timestep of the RNN, we use the current image feature vector x_t to update the vector of hidden variables h_t[3]. We then compute sound features by an affine transformation of the hidden variables:

$$\begin{aligned} \vec{s}_t &= W_{sh} h_t + b_s \\ h_t &= \mathcal{L}(x_t, h_{t-1}) \end{aligned} \tag{3}$$

where $\mathcal{L}$ is a function that updates the hidden state. During training, we minimize the difference between the predicted and ground-truth predictions at each timestep:

$$E(\{\vec{s}_t\}) = \sum_{t=1}^{T} \rho(\|\vec{s}_t - \tilde{\vec{s}}_t\|), \tag{4}$$

where $\tilde{\vec{s}}_t$ and $\vec{s}_t$ are the true and predicted sound features at time t, and $\rho(r) = \log(1 + dr^2)$ is a robust loss that bounds

[2]We use only the first color image to reduce the computational cost of ConvNet features, as subsequent color frames may be redundant with the spacetime images.

[3]For simplicity of presentation, we have omitted the LSTM's hidden cell state, which is also updated at each timestep.

the error at each timestep (we use $d = 25^2$). We also increase robustness of the loss by predicting the square root of the subband envelopes, rather than the envelope values themselves. To make the learning problem easier, we use PCA to project the 42-dimensional feature vector at each timestep down to a 10-dimensional space, and we predict this lower-dimensional vector. When we evaluate the neural network, we invert the PCA transformation to obtain sound features. We train the RNN and CNN jointly using stochastic gradient descent with Caffe [22, 10]. We found it helpful for convergence to remove dropout [39], to clip gradients, and, when training from scratch, to use batch normalization [19]. We also use multiple layers of LSTM (the number depends on the task; see Appendix A2).

5.2. Generating a waveform

We consider two methods for generating a waveform from the predicted sound features. The first is the simple parametric synthesis approach of [28, 37], which iteratively imposes the subband envelopes on a sample of white noise (we used just one iteration). We found that the result can be unnatural for some materials, particularly for hard materials such as wood and metal – perhaps because our predicted sounds lack the fine-grained structure and random variation of real sounds.

Therefore we also consider an example-based synthesis method that snaps a sound prediction to the closest exemplar in the training set. We form a query vector by concatenating the predicted sound features $\vec{s}_1, ..., \vec{s}_T$ (or a subsequence of them), finding its nearest neighbor in the training set as measured by L_1 distance, and transferring its corresponding waveform.

6. Experiments

We applied our sound-prediction model to several tasks, and we evaluated it with human studies and automated metrics.

6.1. Sound prediction tasks

In order to study the problem of detection – that is, the task of determining when and whether an action that produces a sound has occurred – separately from the task of sound prediction, we consider evaluating two kinds of videos. First we focus on the prediction problem and only consider videos centered on amplitude peaks in the ground-truth audio. These peaks largely correspond to impacts, and by centering the sounds this way, we can compare with models that do not have a mechanism to align the audio with the time of the impact (such as those based on nearest-neighbor search with CNN features). To detect these audio peaks, we use a variation of mean shift [12] on the audio amplitude, followed by non-maximal suppression. We then sample a 15-frame sequence (approximately 0.5 seconds) around each detected peak.

For the second task, which we call the *detection-and-prediction* task, we train our models on longer sequences (approximately 2 seconds long) sampled uniformly from the training videos with a 0.5-second stride. We then evaluate the models on full-length videos. Since it is often difficult to discern the precise timing of an impact with subframe accuracy, we allow the predicted features to undergo small shifts before being compared to the ground truth. We also introduce a lag in the RNN output, which allows our model to look a few frames into the future before outputting sound features (see Appendix A2 for more details). For both tasks, we split the full-length videos into a training and test set (75% training and 25% testing).

Models On the centered videos, we compared our model to image-based nearest neighbor search. We computed fc_7 features from a CNN pretrained on ImageNet [25] on the center frame of each sequence, which by construction is the frame where the impact sound occurs. To synthesize sound for a new sequence under this model, we match its center frame to the training set and transfer the sound corresponding to the best match (which is also centered on the middle frame). We considered variations where the CNN features were computed on an RGB image, on (three-frame) spacetime images, and on the concatenation of both features.

We also explored variations of our model to understand the influence of different design decisions. We included models with and without ImageNet pretraining; with and without spacetime images; and with example-based versus parametric waveform generation. Finally, we included a model where the RNN connections were broken (the hidden state was set to zero between timesteps).

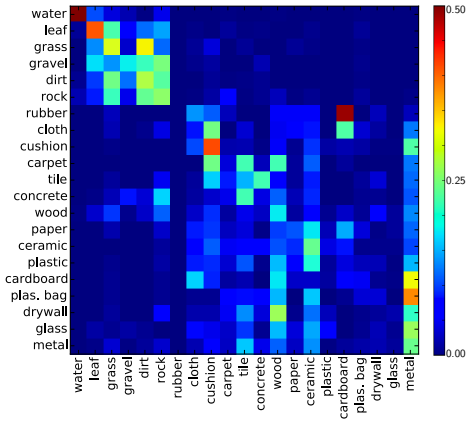
For the RNN models that do example-based waveform generation (Section 5.2), we used the centered impacts in the training set as the exemplar database. For the centered videos we performed the query using the sound features for the entire sequence. For the long videos in the detection-and-prediction task, which contain multiple impact sounds, this is not possible. Instead, we first detect peaks in the amplitude of the parametrically inverted waveform, and match the sound features in a small (8-frame) window beginning one frame before the peak.

6.2. Evaluating the predicted sounds

We would like to assess the quality of the sounds produced by our model, and to understand what the model learned about physics and materials. First, we use automated metrics that measure objective acoustic properties, such as loudness, along with psychophysical experiments to evaluate the plausibility of the sounds to human observers. We then evaluate how effective the predicted sounds are for material and action classification.

Algorithm	Psychophysical study: Labeled *Real*	Loudness: Err.	Loudness: r	Spec. Centroid: Err.	Spec. Centroid: r
Full system	**40.01% ± 1.66**	**0.21**	**0.44**	**3.85**	**0.47**
- Trained from scratch	36.46% ± 1.68	0.24	0.36	4.73	0.33
- No spacetime	**37.88% ± 1.67**	0.22	0.37	4.30	0.37
- Parametric synthesis	34.66% ± 1.62	**0.21**	**0.44**	**3.85**	**0.47**
- No RNN	29.96% ± 1.55	1.24	0.04	7.92	0.28
Image match	32.98% ± 1.59	0.37	0.16	8.39	0.18
Spacetime match	31.92% ± 1.56	0.41	0.14	7.19	0.21
Image + spacetime	33.77% ± 1.58	0.37	0.18	7.74	0.20
Random impact sound	19.77% ± 1.34	0.44	0.00	9.32	0.02

(a) Model evaluation



(b) Predicted sound confusion matrix

Figure 5: (a) We measured the rate that subjects chose an algorithm's synthesized sound over the actual sound. Our full system, which was pretrained from ImageNet and used example-based synthesis to generate a waveform, significantly outperformed models based on image matching. (b) What sounds like what, according to our algorithm? We applied a classifier trained on *real* sounds to the sounds produced by our algorithm to produce a confusion matrix. Rows correspond to confusions made for a single category (*c.f.* Figure 3(b), which shows a confusion matrix for real sounds).

Psychophysical study To test whether the sounds produced by our model varied appropriately with different actions and materials, we conducted a psychophysical study on Amazon Mechanical Turk. We used a two-alternative forced choice test where participants were asked to distinguish between real and fake sounds. We showed them two videos of an impact event – one playing the recorded sound, the other playing a synthesized sound. They were then asked to choose the one that played the real sound. The algorithm used for synthesis was chosen randomly on a per-video basis, along with the order of the two videos. We randomly sampled 15 impact-centered sequences from each full-length video, showing each participant at most one impact from each one. At the start of the experiment we revealed the correct answer to five practice sequences.

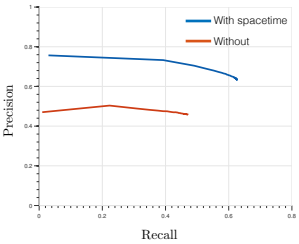
We compared our model to several other methods (Table 5(a)), measuring the rate that participants mistook an algorithm's result for the ground-truth sound. We found that our full system – with RGB and spacetime input, RNN connections, ImageNet pretraining, and example-based waveform generation – significantly outperformed the best image-matching method and a simple baseline where a (centered) sound was chosen at random from the training set ($p < 0.001$ with a two-sided t-test). Our model trained from scratch also significantly outperformed the best image-matching baseline ($p = 0.02$). We did not find the difference between our model with both RGB and spacetime images and RGB-only model on this task to be significant ($p = 0.08$) on the centered videos.

We found that the model in which the RNN connections were broken was often unable to detect the location of the hit, and that it under-predicted the amplitude of the sounds. As a result, it was unable to find good matches, and it performed poorly on automated metrics. The performance of our model with parametric (rather than example-based) waveform generation varied widely between categories. It did well on materials such as *leaves* and *dirt* that are suited to the relatively noisy sounds that the method produces but poorly on hard materials such as *wood* and *metal* (*e.g.* a confusion rate of $63\% \pm 6\%$ for dirt and $19\% \pm 5\%$ for metal).

We show results broken down by semantic category in Figure 7. For some categories (*e.g.* leaves and grass), participants were often fooled by our results; they distinguished

Algorithm	Labeled *Real*
Full sys. + match	40.79% ± 1.66
Full sys.	38.65% ± 1.63
Img. match + match	37.17% ± 1.64
Img. match	31.61% ± 1.59
Random + match	36.00% ± 1.62
Random	20.44% ± 1.38

(a) Oracle experiments



(b) Impact detection

Figure 6: (a) We ran variations of the full system and the image-matching method that used both an RGB and spacetime images. For each model, we include an oracle model that draws its sound examples from videos with the same ground-truth label. (b) Precision-recall curve for impact detection, obtained by detecting hits after rescaling the predicted waveform. Our method with spacetime images outperforms a method with only RGB images.

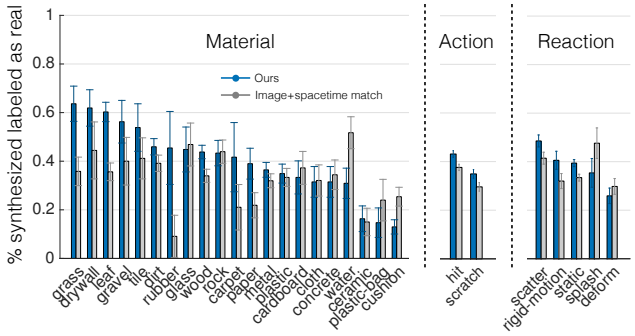


Figure 7: Semantic analysis of psychophysical study. We show the rate that our algorithm fooled human participants for each material, action, and reaction class. The error bars are based on the number of participant responses per category. Our approach significantly outperforms the highest-performing image-matching method (RGB + spacetime).

the real and synthesized sounds at approximately chance levels. For video clips where participants consistently chose our sounds, it may have been because they were more prototypical of the object category. The sound of hitting fallen leaves, for example, is highly varied and may not be fully indicated by the video: we might hear some combination leaves themselves and whatever is underneath them. A generic leaf sound, in many cases, may seem more natural to participants. In contrast, we found that participants were very good at distinguishing real and fake cushion sounds, perhaps because people are sensitive to what they should sound like.

Auditory metrics We measured several quantitative properties of the sounds produced by the centered model. First, we evaluated the loudness of the sound, which we took to be the maximal energy over the full duration of the sound, where we measured energy as the L_2 norm of the (compressed) subband envelopes at each timestep. Second, we compared the sounds' spectral centroids, which we measured by taking the center of mass of the frequency subbands for a one-frame (approximately 0.03 sec.) window around the center of the impact. We found that on both metrics, the network was significantly more accurate than the image-matching methods, both in terms of mean squared error and correlation coefficients (Figure 5(a)).

Oracle results How helpful is material category information? We conducted a second study where we asked how performance would change if we controlled for material-recognition accuracy. Using the subset of the data with material annotations, we created a model that chose a random sound from the same ground-truth category as the input video. We also created a number of oracle models that used these material labels (Figure 6(a)). For the best-performing image-matching model (RGB + spacetime), we restricted the pool of matches to be those with the same label as the input (and similarly for the example-based synthesis method). We found that, while knowing the material was helpful for each method, it was not sufficient, as the oracle models did not outperform our model. In particular, the oracle version of our model significantly outperformed the random-sampling oracle ($p < 10^{-4}$).

Impact detection We also used our methods to produce sounds for long (uncentered) videos, a problem setting that allows us to evaluate their ability to detect impact events. To do this, we generate a waveform from the sound predictions using the parametric method (Section 5.2), and detect amplitude peaks using the method in Section 6.1. We then compare the timing of these amplitude peaks to those of the ground truth, considering an impact to be detected if a predicted spike occurred within 0.1 seconds of it (associating the predicted and ground truth greedily as in [11]). We computed a precision-recall curve using amplitude as a proxy for confidence, rescaling the waveform with different values and running the peak-detection procedure for each gain. In Figure 6(b), we compare our model to one that uses only RGB images, finding that the spacetime images significantly improve the result. We provide qualitative examples in Figure 8, with synthesized videos in the supplementary material.

6.3. Learning about material and action by predicting sounds

By learning to predict sounds, did the network also learn something about material and action? To assess this, we tested whether the network's output sounds are informative about material and action class. We applied the same SVM that was trained to predict material/action class on *real* sound features (Sec. 4) to the sounds predicted by our network. Under this evaluation regime, it is not enough for the network's sounds to merely be distinguishable: they must be close enough to real sounds so as to be classified correctly by an SVM that has never seen a predicted sound. To avoid the influence of pretraining, we used the network that was trained from scratch. We note that this method of evaluation is different from that of recent unsupervised learning techniques [9, 1, 42], which retrain a classifier on an internal representation of the network (*e.g.* fc_7 features), rather than on a ground-truth version of the output.

We applied SVMs for both material class and action class. The resulting confusion matrix for material class is shown in Figure 5(b), with balanced accuracy of 18.2% (this result improves to 23.4% with pretraining; see Appendix A1). This accuracy indicates that our model learned an output representation that is informative about material, even though it was only trained to predict sound. On the task of predicting action class from predicted sounds (again using an SVM classifier trained on real sounds), we are able to

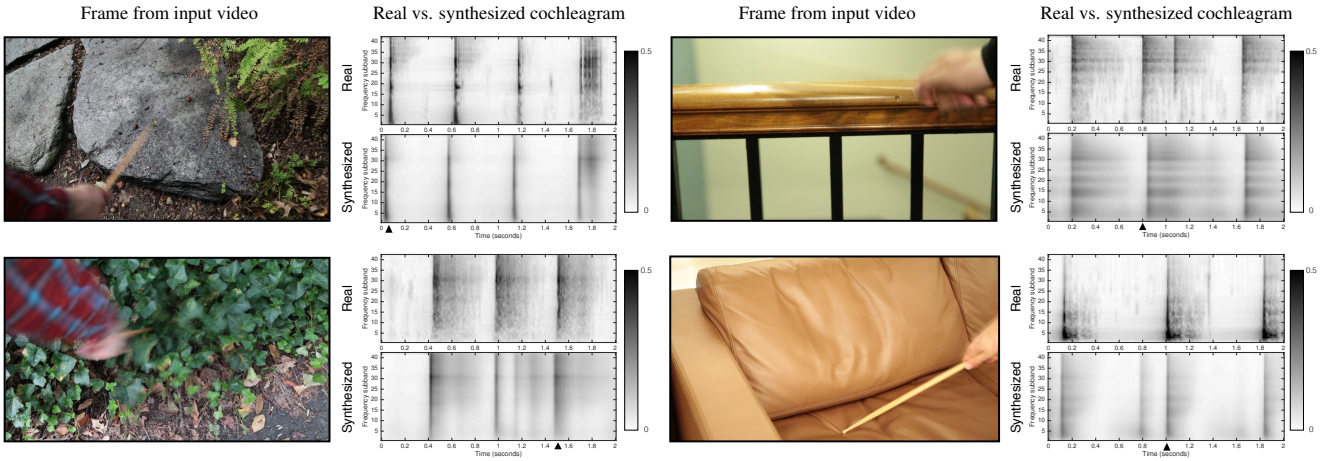


Figure 8: Automatic sound prediction results. We show cochleagrams for a representative selection of video sequences, with a sample frame from each sequence on the left. The frame is sampled from the location indicated by the black triangle on the x-axis of each cochleagram. Notice that the algorithm's synthesized cochleagrams match the general structure of the ground truth cochleagrams. Dark lines in the cochleagrams indicate hits, which the algorithm often detects. The algorithm captures aspects of both the temporal and spectral structure of sounds. It correctly predicts staccato taps in rock example and longer waveforms for rustling ivy. Further, it tends to predict lower pitched thuds for a soft couch and higher pitched clicks when the drumstick hits a hard wooden railing (although the spectral differences may appear small in these visualizations, we evaluate this with objective metrics in Section 6). A common failure mode is that the algorithm misses a hit (railing example) or hallucinates false hits (cushion example). This frequently happens when the drumstick moves erratically. *Please see our supplementary video for qualitative results.*

distinguish hits and scratches with 67.9% class-averaged accuracy (after resampling each class to 2000 examples each). The same classifier gets 84.1% accuracy on real sounds.

Material class confusions are often in the same superordinate category. For example, soft materials, like cloth, are often confused with other soft materials, like cushion, and similarly for hard materials, *e.g.*, tile is often confused with concrete. Quantitatively, the classifier can tell apart hard and soft materials with a balanced accuracy of 69.0% (chance = 50%), where we have defined soft materials to be {*leaf, grass, rubber, cloth cushion, plastic bag*} and hard materials to be {*gravel, rock, tile, concrete, wood, ceramic, plastic, drywall, glass, metal*}.

In Appendix A1, we have also provided a confusion matrix that we obtained by directly predicting material category from visual features (we used pretrained fc_7 CNN features). The kinds of mistakes that this visual classifier made were often different from those of the sound classifier (Figure 3). For instance, the visual classifier was able to distinguish categories that have a very different visual appearance such as *cardboard* and *cushion* – categories that, both being low-pitched sounds, were sometimes are confused by the sound classifier. On the other hand, it was more likely to confuse materials from outdoor scenes, such as rocks and leaves – materials that sound very different but which frequently co-occur in a scene. When we analyze our model by classifying its sound predictions (Figure 5(b)), the resulting confusion matrix contains both kinds of error: *visual analysis* errors when it misidentifies the material that was struck, and *sound synthesis* errors when it produces a sound that was not a convincing replica of the real sound.

7. Discussion

In this work, we proposed the problem of synthesizing visually indicated sounds – a problem that requires an algorithm to learn about material properties and physical interactions. We introduced a dataset for studying this task, which contains videos of a person probing materials in the world with a drumstick, and an algorithm based on recurrent neural networks. We evaluated the quality of our approach with psychophysical experiments and automated metrics, showing that the performance of our algorithm was significantly better than baselines.

We see our work as opening two possible directions for future research. The first is producing realistic sounds from videos, treating sound production as an end in itself. The second direction is to use sound and material interactions as steps toward physical scene understanding. We will release both the *Greatest Hits* dataset and the code for our algorithm.

Acknowledgments. This work was supported by NSF grants 6924450 and 6926677, by Shell, and by a Microsoft Research Fellowship to A.O. We thank Rui Li for the helpful discussions, and the maintenance staffs at Arnold Arboretum and Mt. Auburn Cemetery for not asking too many questions while we were collecting the *Greatest Hits* dataset.

References

[1] P. Agrawal, J. Carreira, and J. Malik. Learning to see by moving. *arXiv preprint arXiv:1505.01596*, 2015. 2, 7

[2] R. Baillargeon. The acquisition of physical knowledge in infancy: A summary in eight lessons. *Blackwell handbook of childhood cognitive development*, 1:46–83, 2002. 1

[3] S. Bell, P. Upchurch, N. Snavely, and K. Bala. Material recognition in the wild with the materials in context database. *CoRR*, abs/1412.0623, 2014. 1

[4] T. Bonebright. Were those coconuts or horse hoofs? visual context effects on identification and perceived veracity of everyday sounds. In *The 18th International Conference on Auditory Display (ICAD2012) Atlanta,(GA)*, 2012. 2

[5] S. Cavaco and M. S. Lewicki. Statistical modeling of intrinsic structures in impacts sounds. *The Journal of the Acoustical Society of America*, 121(6):3558–3568, 2007. 2

[6] A. Davis, K. L. Bouman, M. Rubinstein, F. Durand, and W. T. Freeman. Visual vibrometry: Estimating material properties from small motion in video. In *CVPR*, 2015. 2

[7] A. Davis, M. Rubinstein, N. Wadhwa, G. J. Mysore, F. Durand, and W. T. Freeman. The visual microphone: passive recovery of sound from video. *ACM Transactions on Graphics (TOG)*, 2014. 2

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009. 3

[9] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. *arXiv preprint arXiv:1505.05192*, 2015. 2, 7

[10] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. *CVPR*, 2015. 4, 5

[11] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 7

[12] K. Fukunaga and L. D. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *Information Theory, IEEE Transactions on*, 21(1):32–40, 1975. 5

[13] W. W. Gaver. What in the world do we hear?: An ecological approach to auditory event perception. *Ecological psychology*, 1993. 2

[14] M. Gemici and A. Saxena. Learning haptic representation for manipulating deformable food objects. In *IROS*, 2014. 3

[15] B. R. Glasberg and B. C. Moore. Derivation of auditory filter shapes from notched-noise data. *Hearing research*, 47(1):103–138, 1990. 3

[16] R. Goroshin, J. Bruna, J. Tompson, D. Eigen, and Y. LeCun. Unsupervised feature learning from temporal data. *arXiv preprint arXiv:1504.02518*, 2015. 2

[17] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 4

[18] Y. Hu and P. C. Loizou. Speech enhancement based on wavelet thresholding the multitaper spectrum. *Speech and Audio Processing, IEEE Transactions on*, 12(1):59–67, 2004. 3

[19] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 5, 12

[20] D. Jayaraman and K. Grauman. Learning image representations equivariant to ego-motion. *arXiv preprint arXiv:1505.02206*, 2015. 2

[21] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(1):221–231, 2013. 4

[22] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*, pages 675–678. ACM, 2014. 5

[23] M. Kac. Can one hear the shape of a drum? *American Mathematical Monthly*. 2

[24] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, 2014. 4

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 4, 5, 11, 12

[26] Z.-H. Ling, S.-Y. Kang, H. Zen, A. Senior, M. Schuster, X.-J. Qian, H. M. Meng, and L. Deng. Deep learning for acoustic modeling in parametric speech generation: A systematic review of existing techniques and future trends. *Signal Processing Magazine, IEEE*, 2015. 2

[27] R. A. Lutfi. Human sound source identification. In *Auditory perception of sound sources*, pages 13–42. Springer, 2008. 2

[28] J. H. McDermott and E. P. Simoncelli. Sound texture perception via statistics of the auditory periphery: evidence from sound synthesis. *Neuron*, 71(5):926–940, 2011. 3, 5

[29] H. Mobahi, R. Collobert, and J. Weston. Deep learning from temporal coherence in video. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 737–744. ACM, 2009. 2

[30] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. Multimodal deep learning. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 689–696, 2011. 2

[31] L. Pinto and A. Gupta. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. *arXiv preprint arXiv:1509.06825*, 2015. 3

[32] L. Schulz. The origins of inquiry: Inductive inference and exploration in early childhood. *Trends in cognitive sciences*, 16(7):382–389, 2012. 1

[33] A. A. Shabana. *Theory of vibration: an introduction*, volume 1. Springer Science & Business Media, 1995. 2

[34] L. Sharan, C. Liu, R. Rosenholtz, and E. H. Adelson. Recognizing materials using perceptually inspired features. *International journal of computer vision*, 103(3):348–371, 2013. 1

[35] M. H. Siegel, R. Magid, J. B. Tenenbaum, and L. E. Schulz. Black boxes: Hypothesis testing via indirect perceptual evidence. *Proceedings of the 36th Annual Conference of the Cognitive Science Society*, 2014. 1

[36] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, 2014. 4

[37] M. Slaney. Pattern playback in the 90s. In *NIPS*, pages 827–834, 1994. 3, 5

[38] L. Smith and M. Gasser. The development of embodied cognition: Six lessons from babies. *Artificial life*, 11(1-2):13–29, 2005. 1

[39] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014. 5

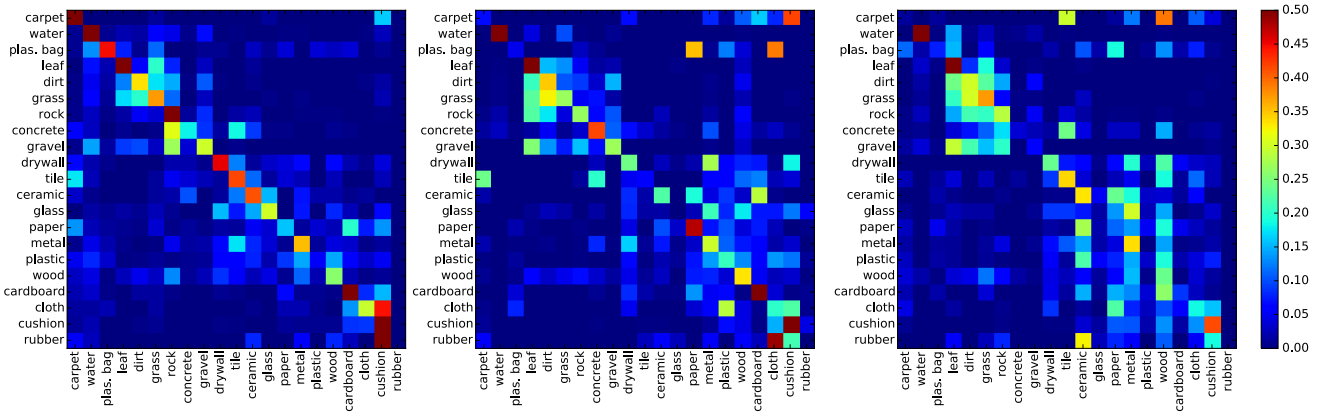
[40] K. Van Den Doel, P. G. Kry, and D. K. Pai. Foleyautomatic: physically-based sound effects for interactive simulation and animation. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 537–544. ACM, 2001. 2

[41] C. Vondrick, H. Pirsiavash, and A. Torralba. Anticipating the future by watching unlabeled video. *arXiv preprint arXiv:1504.08023*, 2015. 2

[42] X. Wang and A. Gupta. Unsupervised learning of visual representations using videos. *arXiv preprint arXiv:1505.00687*, 2015. 2, 7

[43] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *Advances in Neural Information Processing Systems*, 2014. 3

[44] L. Zitnick. 80,000 ms coco images in 5 minutes. In *https://www.youtube.com/watch?v=ZUIEOUoCLBo*. 12



(a) Sound confusion (b) Visual confusion (c) Predicted sound confusion (pretrained model)

Figure A1: For comparison, we have provided confusion matrices for different methods. (a) An SVM trained on subband envelope features and (b) SVM trained on *fc7* features. (c) An SVM trained on *real* sounds (*i.e.* (a)) applied to the sounds produced by our algorithm. In Figure 5(b), we give the predicted sound confusion matrix for a network trained from scratch. We sorted the labels by clustering the rows of the sound confusion matrix.

A1. Material confusion matrices

In Figure A1, we give confusion matrices for material classification with sound features and vision features (AlexNet fc_7 features). We also classify the sound features predicted by our algorithm using a classifier trained with real sounds, as in Figure 5(b). For consistency with the fc_7 model, we used a network pretrained for ImageNet classification [25] (rather than the model that was trained from scratch, as in Figure 5(b)). The convolutional layers in this model were not changed from the original pretrained model; only the fully connected layers. We resample each category to at most 300 examples. The sound classifier's balanced accuracy was 40.0%; the visual classifier's was 30.2%; and the predicted-sound classifier's was 23.4% (18.2% for the model trained from scratch).

A2. Implementation

A2.1. Detection-and-prediction model

We describe our model for the detection-and-prediction task (Section 6.1) in more detail.

Timing We allow the sound features to undergo small time shifts in order to account for misalignments for the detection-and-prediction task. During each iteration of backpropagation, we shift the sequence so as to minimize the loss in Equation 4. We resample the feature predictions to create a new sequence $\hat{\vec{s}}_1, \hat{\vec{s}}_2, ..., \hat{\vec{s}}_T$ such that $\hat{\vec{s}}_t = \vec{s}_{t+L_t}$ for some small shift L_t (we use a maximum shift of 8 samples, approximately 0.09 seconds). During each iteration, we infer this shift by finding the optimal labeling of a Hidden Markov Model:

$$\sum_{t=1}^{T} w_t \rho(\|\hat{\vec{s}}_t - \tilde{\vec{s}}_t\|) + V(L_t, L_{t+1}), \tag{5}$$

where V is a smoothness term for neighboring shifts. For this, we use a Potts model weighted by $\frac{1}{2}(\|\tilde{\vec{s}}_t\| + \|\tilde{\vec{s}}_{t+1}\|)$ to discourage the model from shifting the sound near high-amplitude regions. We also include a weight variable $w_t = 1 + \alpha\delta(\tau \leq ||\tilde{\vec{s}}_t||)$ to decrease the importance of silent portions of the video (we use $\alpha = 3$ and $\tau = 2.2$). During each iteration of backpropagation, we align the two sequences, then propagate the gradients of the loss to the shifted sequence.

To give the RNN more temporal context for its predictions, we also delay its predictions, so that at frame f, it predicts the sound features for frame $f - 2$.

Transforming features for neighbor search For the detection-and-prediction task, the statistics of the synthesized sound features can differ significantly from those of the ground truth – for example, we found the amplitude of peaks in the predicted waveforms to be smaller than those of real sounds. We correct for these differences during example-based synthesis (Section 5.2) by applying a coloring transformation before the nearest-neighbor search. More specifically, we obtain a whitening transformation for the predicted sound features by running the neural network on the test videos and estimating the empirical mean and covariance at the detected amplitude peaks, discarding peaks whose amplitude is below a threshold. We then estimate a similar transformation for ground-truth amplitude peaks in the training set, and we use these transformations to color (*i.e.* transform the mean and covariance of) the predicted features into the space of real features before computing

their L_1 nearest neighbors. To avoid the influence of multiple, overlapping impacts on the nearest neighbor search, we use a search window that starts at the beginning fo the amplitude spike.

Evaluating the RNN for long videos When evaluating our model on long videos, we run the RNN on 10-second subsequences that overlap by 30%, transitioning between consecutive predictions at the time that has the least sum-of-squares difference between the overlapping predictions.

A2.2. Network structure

We used AlexNet [25] for our CNN architecture. For the pretrained models, we precomputed the *pool5* features and fine-tuned the model's two fully-connected layers. For the model that was trained from scratch, we applied batch normalization [19] to each training mini-batch. For the centered videos, we used two LSTM layers with a 256-dimensional hidden state (and three for the detection-and-prediction model). When using multiple LSTM layers, we compensate for the difference in video and audio sampling rates by upsampling the input to the last LSTM layer (rather than upsampling the CNN features), replicating each input k times (where again $k = 3$).

A3. Dataset

In Figure A2, we show a "walk" through the dataset using *fc7* features, similar to [44]. Our data was collected using a wooden (hickory) drumstick, and an SLR camera with a 29.97 Hz framerate. The drumstick hits were performed by the authors. Online workers labeled the impacts by visually examining silent videos. To measure consistency between workers, we labeled a subset of the impacts with 3 or more workers, finding that their material labels agreed with the majority 87.6% of the time. Common inconsistencies include confusing *dirt* with *leaf* (confused 5% of the time); *grass* with *dirt* and *leaf* (8% each); *cloth* with *cushion* (9% of the time).

Figure A2: A "walk" through the dataset using AlexNet *fc7* nearest-neighbor matches. Starting from the left, we matched an image with the database and placed its best match to its right. We repeat this 5 times, with 20 random initializations. We used only images taken at a contact point (the middle frames from the "centered" videos). To avoid loops, we removed videos when any of their images were matched. The location of the hit, material, and action often vary during the walk. In some sequences, the arm is the dominant feature that is matched between scenes.