Material Converter: Manipulating Materials of Visual Objects with Sound

Tingle Li[1,3], Yichen Liu[1], Andrew Owens[2], Hang Zhao[1,3]
[1]IIIS, Tsinghua University [2]University of Michigan [3]Shanghai Qi Zhi Institute

1. Introduction

Today's image synthesis methods [7, 13] largely rely on humans to specify their tasks. While recent work has increased the flexibility of these systems by incorporating natural language instructions [1, 8, 11, 12], it can be challenging for language to capture important nuance. Audio, by contrast, can often convey important distinctions that would be ambiguous in language alone. For example, asking a model to generate a "footstep in mud" is highly ambiguous; specifying the sound of the footstep, on the other hand, conveys whether the mud is deep or shallow, how wet or dry it is, and the force of the footstep.

In this work, we propose to translate the texture of visual objects in a scene to a new material, given only an *impact sound* that specifies how the new material should sound when it is struck. We call this task *audio-visual material conversion* (Figure 1). Impact sounds are produced by the interaction between different objects, and are highly dependent on the materials of the objects and the forces involved. They therefore provide useful information about material properties and forces. The resulting video should retain the structure of the original scene, while converting the texture of the material to that of the sound.

To this end, our proposed method, *i.e.*, Material Converter, contains two distinct training objectives for converting texture and preserving structure, respectively. The first objective involves a GAN loss that converts the texture of a source video. The second objective leverages contrastive learning to maximize the mutual information between the source and converted videos, which aims to preserve the object structure after conversion. These two objectives can be jointly applied for end-to-end training.

Furthermore, the target material sound serves as an audio-cue that controls the network to generate the corresponding material texture. Since sounds of different materials are provided during training, our Material Converter is able to convert object texture from one material to many other materials based on the given sounds.

2. Material Converter

Regarding the task of audio-visual material conversion, the general goal is to learn a video[1] feature mapping from the source video domain $\mathcal{X}$ to the target video domain $\mathcal{Y}$, where $\mathcal{Y}$ is determined by the sound (audio cues) produced by various materials, denoted as the audio domain $\mathcal{A}$. To achieve this goal, we propose an unsupervised training method called Material Converter, which can be trained on unpaired video samples, *i.e.*, the video pairs taken from $\mathcal{X}$ and $\mathcal{Y}$ are not sharing content. This can be accomplished through two distinct training objectives which will be described in this section.

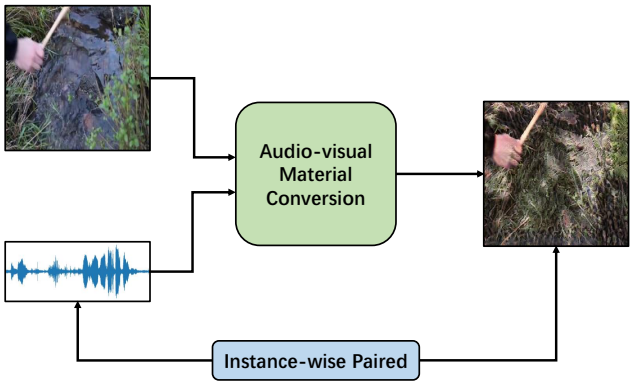

Figure 1. The schematic diagram of the audio-visual material conversion task. Given a source video and target material audio as input, this task aims to convert the source video to a new video that corresponds to the target material audio. This is an example where water is converted to grass.

2.1. Texture Conversion via Adversarial Training

A popular solution for unpaired image texture conversion is CycleGAN [13]. In contrast to CycleGAN that leverages two GANs, our Material Converter only requires a single GAN, which largely simplifies the training process. Specifically, the generator network G can be divided into two components, an encoder G_{enc} followed by a decoder G_{dec}. For a given dataset of unpaired video instances $X = \{\boldsymbol{x} \in \mathcal{X}\}$, $Y = \{\boldsymbol{y} \in \mathcal{Y}\}$, and the audio cues $A_Y = \{\boldsymbol{a}_Y \in \mathcal{A}\}$ related to Y, G_{enc} and G_{dec} are applied sequentially to generate the output video $\hat{\boldsymbol{y}} = G_{\text{dec}}(\text{concat}(G_{\text{enc}}(\boldsymbol{x}), \boldsymbol{a}_Y))$. An adversarial loss [4] is then applied to encourage $\hat{\boldsymbol{y}}$ to approach the visual features, *i.e.*, the texture, of the target domain $\mathcal{Y}$ under the guidance of $\boldsymbol{a}_Y$:

$$\begin{aligned}\mathcal{L}_{\text{GAN}}(G_{X\to Y}, D_Y) = \; & \mathbb{E}_{\boldsymbol{y}\sim Y} \log D(\boldsymbol{y}, \boldsymbol{a}_Y) + \\ & \mathbb{E}_{\boldsymbol{x}\sim X} \log\left(1 - D(G(\boldsymbol{x}, \boldsymbol{a}_Y), \boldsymbol{a}_Y)\right)\end{aligned} \tag{1}$$

where D is the discriminator network. Please note that the fusion of two modalities in D is an early fusion, where

[1]To avoid confusion, the word "video" here refers to the visual modality only, *i.e.*, stack of frames, not the idiomatic meaning of a video which involves both the visual and audio contents.

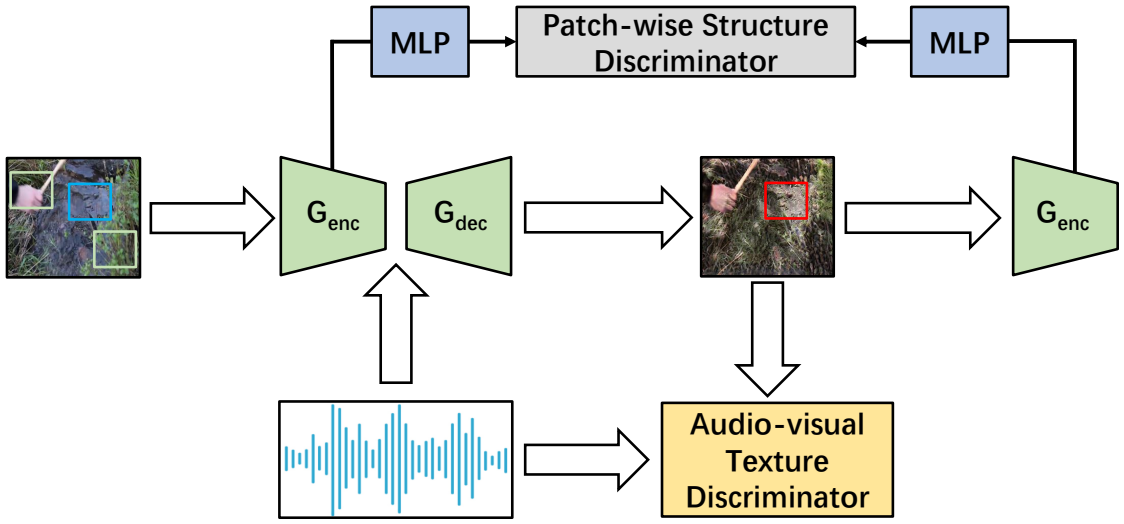

Figure 2. An overview of the architecture of Material Converter. The patch-wise structure discriminator [10] is used to preserve object structure, while the audio-visual texture discriminator is used to maintain object texture. This is an example where water puddle is converted to grass lawn. The **generated grass patch** should match its corresponding **input water patch**, in comparison to **other random patches**. Note that the MLP component has been ignored during inference.

the spectrogram feature of $\boldsymbol{a}_Y$ is concatenated to $\hat{\boldsymbol{y}} = G(\boldsymbol{x}, \boldsymbol{a}_Y)$ before feeding into D. We empirically found that such early fusion yields better results in terms of video quality.

2.2. Preserving Object Structure via Contrastive Learning

In this task, a successfully synthesized video should be equipped with the material texture of the target video while fully preserving the structure of the source video. However, both information, *i.e.*, texture and structure information, are inherently entangled within the learned feature, and adversarial training can only guarantee texture transfer. One trivial solution could be that we get the same video for any inputs. Therefore, as shown in Figure 2, we introduce the second training objective, based on noise contrastive estimation (NCE) [5], which aims to preserve structure information by establishing mutual correspondence between the source and generated videos, $\boldsymbol{x}$ and $\hat{\boldsymbol{y}}$ respectively. Note that this training objective is only employed to the encoder network G_{enc}, which is a multi-layer convolutional network that transforms the source video into feature stacks at each layer. In this way, we encourage G_{enc} to abandon the texture of the source material while preserving the structure of the source video; then the job of the decoder network G_{dec} is to add to the video the target material texture.

Given a "query" vector $\boldsymbol{q}$, the fundamental objective in contrastive learning is to optimize the probability of selecting the corresponding "positive" sample $\boldsymbol{v}^+$ among N "negative" samples $\boldsymbol{v}^-$. The query, positive and N negatives are transformed to M-dimensional vectors, *i.e.*, $\boldsymbol{q}, \boldsymbol{v}^+ \in \mathbb{R}^M$ and $\boldsymbol{v}^- \in \mathbb{R}^{N \times M}$. This problem setting can be expressed as a multi-classification task with $N+1$ classes:

$$\ell(\boldsymbol{q}, \boldsymbol{v}^+, \boldsymbol{v}^-) = -\log\left(\frac{\exp(\frac{\boldsymbol{q}\cdot\boldsymbol{v}^+}{\tau})}{\exp(\frac{\boldsymbol{q}\cdot\boldsymbol{v}^+}{\tau}) + \Sigma_{n=1}^{N}\exp(\frac{\boldsymbol{q}\cdot\boldsymbol{v}_n^-}{\tau})}\right) \quad (2)$$

where $\boldsymbol{v}_n^-$ denotes the n-th negative sample and τ is a temperature parameter, as suggested in SimCLR [2], that scales the similarity distance between $\boldsymbol{q}$ and other samples. The crossentropy term in 2 represents the probability of matching $\boldsymbol{q}$ with the corresponding positive sample $\boldsymbol{v}^+$. Thus, iteratively minimizing the negative log-crossentropy is equivalent to establishing mutual correspondence between the query space and the sample space.

In our task, we draw the $N+1$ positive/negative samples from the the source video $\boldsymbol{x} \in X$, and the query $\boldsymbol{q}$ is selected from the generated video $\hat{\boldsymbol{y}}$. From Figure 2, it can be seen that the selected samples are "patches" that capture local information among the video features. This setup is motivated by the logical assumption that the global correspondence between $\boldsymbol{x}$ and $\hat{\boldsymbol{y}}$ is determined by the local, *i.e.*, patch-wise, correspondences.

Since the encoder G_{enc} is a multi-layer convolutional network that maps $\boldsymbol{x}$ into feature stacks after each layer, we choose L layers and pass their feature stacks through a small MLP network P. The output of P is $P(G_{\text{enc}}^l(\boldsymbol{x})) = \{\boldsymbol{v}_l^1, ..., \boldsymbol{v}_l^N, \boldsymbol{v}_l^{N+1}\}$, where $l \in \{1, 2..., L\}$ denotes the index of the chosen encoder layers and $G_{\text{enc}}^l(\boldsymbol{x})$ is the output feature stack of the l-th layer. Similarly, we can obtain the query set by encoding the generated spectrogram $\hat{\boldsymbol{y}}$ into $\{\boldsymbol{q}_l^1, ..., \boldsymbol{q}_l^N, \boldsymbol{q}_l^{N+1}\} = P(G_{\text{enc}}^l(\hat{\boldsymbol{y}}))$. Now we let $\boldsymbol{v}_l^n \in \mathbb{R}^M$ and $\boldsymbol{v}_l^{(N+1)\setminus n} \in \mathbb{R}^{N \times M}$ denote the corresponding positive sample and the N negative samples, respectively, where n

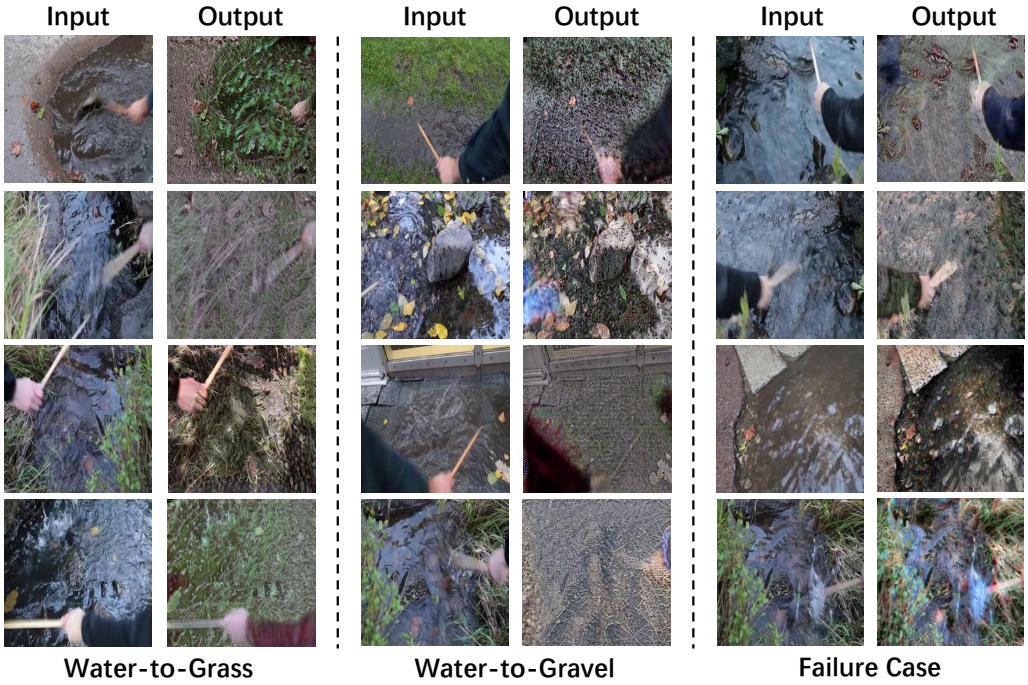

Figure 3. Quality results on the Greatest Hits dataset. Both input and output are extracted with 1 frame from videos for better presentation. Since there are no baseline methods, we only show successful cases generated by Material Converter, namely water→grass and water→gravel, in the first two categories. And the last category present some failure cases.

is the sample index and M is the channel size of P. By referring to Eq. (2), our second training objective can be expressed as:

$$\mathcal{L}_{\text{NCE}}(G_{\text{enc}}, P, X) = \mathbb{E}_{\boldsymbol{x}\sim X} \sum_{l=1}^{L} \sum_{n=1}^{N+1} \ell(\boldsymbol{q}_l^n, \boldsymbol{v}_l^n, \boldsymbol{v}_l^{(N+1)\setminus n}) \tag{3}$$

which is the average NCE loss from all L encoder layers.

2.3. Overall Objective

In addition to the two objectives discussed above, we have also employed an identity loss $\mathcal{L}_{\text{identity}} = \mathcal{L}_{\text{NCE}}(G_{\text{enc}}, P, Y)$ which utilizes the NCE expression in Eq. (3). By taking the NCE loss on the identity generation process, *i.e.*, generating $\hat{\boldsymbol{y}}$ from $\boldsymbol{y}$, we are likely to prevent the generator from making unexpected changes. Now we can define our final training objective as:

$$\begin{aligned}\mathcal{L}_{\text{final}} = \; & \mathcal{L}_{\text{GAN}}(G_{X\to Y}, D_Y) + \lambda \mathcal{L}_{\text{NCE}}(G_{\text{enc}}, P, X) + \\ & \mu \mathcal{L}_{\text{NCE}}(G_{\text{enc}}, P, Y)\end{aligned} \tag{4}$$

where λ and μ are two parameters for adjusting the strengths of the NCE and the identity loss.

2.4. Experiment

2.5. Experimental Setup

Dataset. The Greatest Hits dataset [9], which contains a drumstick hitting, scratching, and poking different objects in both indoor and outdoor scenes, was used to evaluate our proposed model. Specifically, there are 977 videos from indoor (64%) and outdoor scenes (36%). However, since this dataset was first collected for the sound generation task, each video is somehow distinctive from the background scene, making it difficult to use for audio-visual material conversion task directly. For example, dirt videos are often accompanied with flowers, leaves, grass and gravels, spawning it difficult to identify the objects that need to be converted. To mitigate this effect, we manually select videos from 3 material classes, *i.e.*, water, gravel and grass, where the backgrounds are less diverse. Besides, collecting more suitable data is a plan for our future direction.

Network architecture. The encoder and decoder of the GAN generator are fully 3D convolutional networks, with 6 layers of ResNet-based CNN bottlenecks [6] in between. The kernel size is set to $3\times3\times3$ for all CNNs, and the stride size depends on whether downsampling is required. While for the discriminator, we applied the PatchGAN architecture [7] as used in CycleGAN [13]. Besides, a ResNet18 [6] backbone is employed to extract features before feeding to the decoder of the GAN generator. In addition, before computing the contrastive loss, we first take intermediate features with five different scales from the generator's encoder, then a two-layer MLP with 256 units is applied for each selected feature.

Training setting. We design a pre-processing paradigm for training efficiency as follows: (1) each video is interpolated to 256×256 and sampled 32 frames uniformly before saving as images; (2) each audio is randomly cropped or tiled to a fixed duration of 3s first, then converted to 16 kHz and 32-bit precision in floating-point PCM format. Finally, a 512-point discrete Fourier transform is performed using nnAudio [3], with 25 ms frame length and 10 ms frame-shift. Please note that only the magnitude spectrogram is taken to the decoder of the GAN and the audio backbone. When it comes to the hyper-parameters, our proposed method was trained with a batch size of 1 and an initial learning rate of 2e-4 for 200 epochs, using the Adam optimizer. We only use random horizontal flip as the video data augmentation. Other training schemes are kept the same as those in the official implementation of CUT [10].

2.6. Results

Regarding this generative task, we have provided qualitative results here for a subjective evaluation, as presented in Figure 3. Note that this is achieved by only a single Material Converter that was given audios of different materials, whose goal is to encourage the model to deduce a texture choice for the corresponding material. Specifically, judging from the first two categories that belong to water→grass and water→gravel respectively, our Material Converter can generate consistent videos, and gains the capability of one-to-many conversion under the guidance of the audio cues. However, the texture of the human arm and hand in Figure 3 are also converted towards the target material, suggesting that our Material Converter has yet to possess the ability of finding the object that needs to be converted from the video. In other words, it drastically converts the entire scene and we aim to solve this issue in future work.

Furthermore, the failure cases, which are mainly caused by the indistinguishable material audio cues, are shown in the last category in Figure 3. Interestingly, it turns out that the generated videos are somewhat contrived, manifesting that our Material Converter is prone to collapse when giving ambiguous audio. Hence, it can be stated that the audio cues matter a lot in terms of the conversion process.

3. Conclusion

In this paper, we introduce a novel task, *i.e.*, audio-visual material conversion, which aims to convert object material texture corresponding to a given audio. To tackle this task, we propose Material Converter, a contrastive-based audio-visual GAN model, which can convert materials of visual objects with audio cues. Experimental results on the Greatest Hits dataset show that our Material Converter is able to generate consistent videos given distinguishable material audio. However, Material Converter will sometimes collapse by giving ambiguous material audio. We hope our work will shed new light on the cross-modal controllable image-to-image translation field.

References

[1] D. Bau, A. Andonian, A. Cui, Y. Park, A. Jahanian, A. Oliva, and A. Torralba. Paint by word. In *arXiv:2103.10951*, 2021. 1

[2] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607, 2020. 2

[3] K. W. Cheuk, H. Anderson, K. Agres, and D. Herremans. nnAudio: An on-the-fly gpu audio to spectrogram conversion toolbox using 1d convolutional neural networks. *IEEE Access*, 8:161981–162003, 2020. 4

[4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014. 1

[5] M. Gutmann and A. Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 297–304, 2010. 2

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[7] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 1, 3

[8] J. Johnson, A. Gupta, and L. Fei-Fei. Image generation from scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1219–1228, 2018. 1

[9] A. Owens, P. Isola, J. McDermott, A. Torralba, E. H. Adelson, and W. T. Freeman. Visually indicated sounds. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2405–2413, 2016. 3

[10] T. Park, A. A. Efros, R. Zhang, and J.-Y. Zhu. Contrastive learning for unpaired image-to-image translation. In *European Conference on Computer Vision*, pages 319–345, 2020. 2, 4

[11] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever. Zero-shot text-to-image generation. *arXiv preprint arXiv:2102.12092*, 2021. 1

[12] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. In *International Conference on Machine Learning*, pages 1060–1069, 2016. 1

[13] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2223–2232, 2017. 1, 3