Automated Conversion of Music Videos into Lyric Videos

Jiaju Ma
Stanford University

Anyi Rao
Stanford University

Li-Yi Wei
Adobe Research

Rubaiat Habib Kazi
Adobe Research

Valentina Shin
Adobe Research

Maneesh Agrawala
Stanford University
Roblox

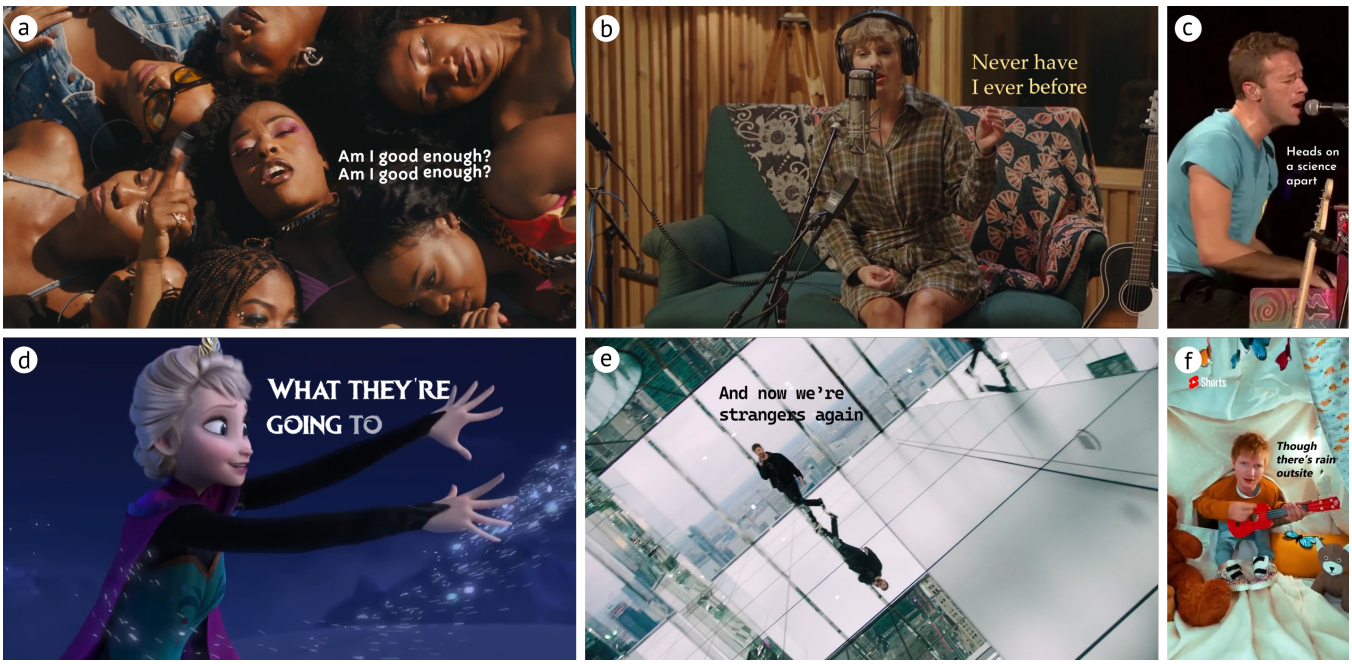

Figure 1: We propose a set of design guidelines for adding lyrics to music videos in a manner that ensures text readability and unifies the viewer's focus of attention. We further implement a fully automated pipeline that instantiates these guidelines to convert an input music video into a lyric video. The results shown above demonstrate that our pipeline is able to generate lyric videos from a wide variety of inputs. ***Video source: (a) Selfish Soul by Sudan Archives [3], (b) August by Taylor Swift ©Taylor Swift [45], (c) The Scientist by Coldplay [10], (d) Let It Go by Idina Menzel ©2013 Hollywood Records, Inc. [28], (e) iPad by Chainsmokers [7], (f) Sandman by Ed Sheeran [44].*** **(a, b, d, e) are in landscape format while (c, f) are in portrait format.**

ABSTRACT

Musicians and fans often produce lyric videos, a form of music videos that showcase the song's lyrics, for their favorite songs. However, making such videos can be challenging and time-consuming as the lyrics need to be added in synchrony and visual harmony with the video. Informed by prior work and close examination of existing lyric videos, we propose a set of design guidelines to help creators make such videos. Our guidelines ensure the readability of the lyric text while maintaining a unified focus of attention. We instantiate these guidelines in a fully automated pipeline that converts an input music video into a lyric video. We demonstrate the robustness of our pipeline by generating lyric videos from a diverse range of input sources. A user study shows that lyric videos generated by our pipeline are effective in maintaining text readability and unifying the focus of attention.

UIST '23, October 29–November 01, 2023, San Francisco, CA, USA

ACM ISBN 979-8-4007-0132-0/23/10…$15.00
https://doi.org/10.1145/3586183.3606757

CCS CONCEPTS

• **Human-centered computing → Human computer interaction (HCI)**.

KEYWORDS

Design guidelines; video generation; lyrics

ACM Reference Format:
Jiaju Ma, Anyi Rao, Li-Yi Wei, Rubaiat Habib Kazi, Valentina Shin, and Maneesh Agrawala. 2023. Automated Conversion of Music Videos into Lyric Videos. In *The 36th Annual ACM Symposium on User Interface Software and Technology (UIST '23), October 29–November 01, 2023, San Francisco, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3586183.3606757

1 INTRODUCTION

A lyric video is a music video that displays the lyrics with the video imagery. The history of lyric videos can be traced back to 1965, when Bob Dylan released a video for his song *Subterranean Homesick Blues* in which he flips through a pile of cards with words from the song written on them [13]. Following his lead, artists start to release lyric videos, such as Prince [36], Katy Perry [35], Taylor Swift [46], and many more. Other than featuring the music, lyric videos have many other use cases like lip sync and karaoke. Moreover, the presence of text allows the video content to be consumed without audio, either in noisy environments or converted to other media formats like images and GIFs. One can find many examples of this on platforms like Pinterest and Reddit [40]. For these reasons, music fans often make lyric videos themselves by adding animated lyric text to existing music videos, giving birth to YouTube channels like HQG Studios [54] whose videos have gathered millions of views. Social media apps like Instagram and TikTok also have functionalities to help users display song lyrics in their videos.

However, making a lyric video from a music video remains challenging and time-consuming as it requires delicate coordination of audio, visual, and text content [21, 49, 55]. First, the creator needs to ensure the readability of the lyric text. This entails segmenting the body of text into phrases to be shown in the video sequentially and deliberately adding line breaks when the text is long. Second, since the added text requires the viewer's attention to read and process, it needs to be in synchrony and visual harmony with the song and video to minimize distractions and unify the viewer's focus. Achieving these goals requires synchronizing the lyrics to the song and coordinating the text's placement with the video imagery.

Prior work has proposed automated solutions to add text to videos in other forms like subtitles [18], kinetic typography [21, 50], and data visualizations [47]. However, these works only partially considered the text readability and the coordination of text, audio, and video, but these intertwined challenges need to be taken into account holistically. For example, changing the text of a lyric phrase affects when it should appear temporally, as well as its optimal position within the video frame since its size may also change.

To help creators make lyric videos that ensure good readability of the text and maintain the viewer's focus of attention, we propose a set of design guidelines formulated by analyzing guidelines and popular lyric videos. To further assist the creators and validate our proposed design guidelines, we implement a fully automated pipeline that instantiates these design guidelines to convert an input music video to a lyric video. The user can optionally specify the font, color, size, and animation of the lyric texts and adjust algorithm parameters to fine tune the text layout and placement.

We demonstrate the efficacy of our automated pipeline by presenting a wide variety of auto-generated example videos (Figure 1 and Figure 6)[1]. We further evaluate our design guidelines and pipeline through a user study in which 57 participants rated four variations of a lyric video. The results show that lyric videos generated by our pipeline are significantly better at maintaining text readability and unifying viewer attention. In summary, our work makes the following contributions:

(1) A set of design guidelines for making lyric videos that ensure text readability and unify the viewer's focus of attention.
(2) A fully automated pipeline that instantiates these design guidelines to produce lyric videos from input music videos, which can be any video with a song as the background music.

2 RELATED WORK

2.1 Lyric Phrase Content and Layout

We use the term *lyric phrase* to refer to a group of lyric text that appears in the video between specific in and out times. No work has directly studied lyric text readability in videos, so we look to prior work [2, 12, 30, 34, 53, 55] on the readability of video subtitles (transcriptions of spoken words that appear at the bottom of video frame). These works suggest that long lines of text can be hard to read, so a body of text should be segmented into phrases with appropriate line breaks for readability. Popular tools like Adobe Premiere Pro [20] and YouTube Studio [26] use simple rule-based approaches to split a body of transcriptions into subtitle phrases based on either punctuation, character length, or temporal duration.

Unlike regular speeches, songs are composed of vocal phrases that group a series of lyric words together. Given this structure, our approach automatically organizes a song's lyrics into lyric phrases based on the temporal proximity of sequential words. Once a phrase has been determined, text segmentation considers where line breaks should be inserted to divide the text into more lines. Some studies suggest that breaking at linguistic units, such as clauses and sentences, results in better user preference [15, 34]. This is not applicable to our work as song lyrics often "do not follow formal standards of written text composition and lack punctuation" [48]. We instead break longer text into two or more lines where each line has consistent length to minimize the amount of eye movements [34], as excessive eye movements can distract the viewer and result in eyestrain [18]. Doing so also conforms to the graphic design principle of leaving no "runt" in new lines [4].

2.2 Text Placement Near Focus of Attention

Because the human eye can only read text within a small vision span [39], a set of work places text near but not occluding objects that are under the viewer's focus of attention. In subtitling, dynamic subtitles [6] refer to subtitles placed near the speakers or other salient regions. Hu et al. [18] detect where the current speaker is and puts the subtitles near there. A View on the Viewer [24] adjusts the subtitle locations live based on the viewer's eye gaze. Kurzhals et al. [23] show that dynamic subtitles reduce the amount of eye movements and help keep the viewer's attention closer to the image content. Brown et al. [6] similarly find that dynamic subtitles allowed the viewers to miss less of the video content and pick up

[1] Our results can be viewed at https://hci.stanford.edu/research/lyricvideo/

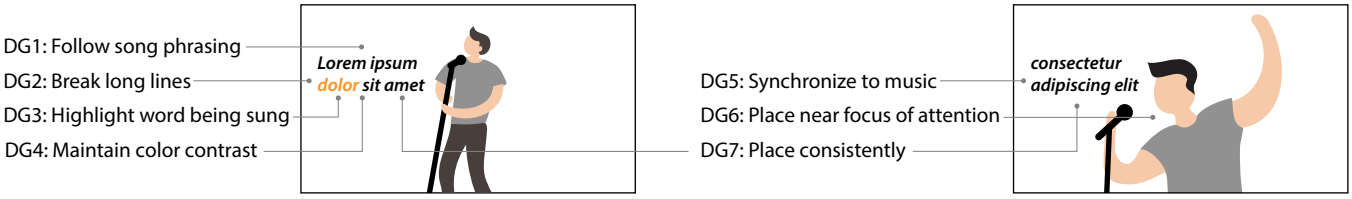

Figure 2: *Illustration of the design guidelines for text readability and unified attention.* A lyric phrase should consist of words closely sung together (DG1), and a long text line should be broken into shorter ones consistent in length (DG2). The word currently being sung should be highlighted (DG3). Text should be placed in areas with sufficient color contrast (DG4) and near the viewer's focus of attention (DG6). Sequential phrases should be placed in similar places (DG7). All lyric phrases should be synchronized to the song (DG5).

more non-verbal cues. In other related domains, SmartShots [47] positions data visualizations near objects they are referencing. SmartOverlays [17] places text labels near salient regions of objects detected by computer vision models to help user identify them.

Following this body of work, we devise a design guideline on placing lyric phrases near but not occluding objects under the focus of attention and use a combination of object detection and segmentation models in our pipeline to instantiate this.

2.3 Tools for Adding Text to Videos

Prior work has built tools to assist with the workflow of adding text to videos. Wang et al. [50] propose an automated framework that visualizes nonverbal sounds in video with onomatopoeias (e.g. "vroom" for engine sounds). The size and opacity of the added sound words are animated by the sound volume. However, visualizing single words means that the system does not need to choose what words should be displayed together and their layout. Moreover, the output video is given as-is and cannot be further edited.

EnACT [49] is a manual tool for adding animated captions to videos. The user can annotate words in the input transcription with emotions. EnACT then overlays the transcription as captions on the input video and applies predefined animations to the annotated texts. This tool can be used to create lyric videos like ours, but the creation process remains manual, such as timing each word to the song and placing the text in the video frame.

TextAlive [21] is a design tool for making kinetic typography videos in which lyrics are animated in synchrony with the song. Similar to our work, it automatically aligns every word in the lyrics to the song and assigns default animations to them. The user can edit the animation further manually. However, the focus of this tools is on kinetic typography videos in which text takes the center role and does not need to accompany any underlying video content, unlike lyric videos where the coordination between text and video imagery is crucial for readability and focus of attention.

3 DESIGN GUIDELINES

We follow the methodology by Agrawala et al. [1] to identify lyric video design guidelines (**DG**) from instructions, examples, and prior work (Figure 2). We first analyzed 15 text tutorials, 5 video tutorials, and 3 subtitle guidelines [12, 53, 55] to form draft guidelines if similar content appeared repeatedly. While a few of these guidelines were concrete (**DG4** and **DG5**), others remained vague at this stage. For example, **DG1** and **DG2** were about "having some words in one or two lines." **DG6** was "don't obstruct video content."

To formalize the design guidelines, we analyzed the top 100 most viewed lyric videos to find patterns (full video list in supplemental materials). For example, **DG7** is a recurring observation on text placement. We also observed common highlighting animations to use as **DG3**'s default options. Prior work also contributed to the guideline definitions. Reducing eye movements [18, 23, 34] helped define **DG2**, **DG6**, and **DG7**. Graphic design layout principle [4] contributed to **DG2**, and sheet music composition to **DG1**.

We present the set of design guidelines below. The two overarching goals of our guidelines are (1) ensuring the readability of the added lyric text and (2) maintaining a unified focus of attention.

3.1 Text Readability

Good text readability can be achieved by properly composing words into lyric phrases with line breaks. Moreover, presenting the text with sufficient contrast and animated highlighting can help guide the viewer's eyes to quickly locate the right words.

DG1: A lyric phrase should consist of words sung closely together. In sheet music, a phrase mark (slur) spans over a set of notes to indicate that they should be sung together as a phrase. These musical phrases are the building blocks of a song. Therefore, a lyric phrase should respect such phrasing by incorporating words in the same musical phrase and display them in the video as a unit.

DG2: Long text in a lyric phrase should be broken into two or more lines with consistent length. Some lyric phrases might contain many words, such as in a fast-paced song. A long line of text is slower and harder to read because it requires excessive eye movement; it can also be distracting when added into the video, as suggested by prior research [2]. Because of this, a long line should be broken in two or more lines with consistent length. Doing so minimizes the amount of eye movement when reading from line to line [34] and also conforms to the graphic design principle of leaving no "runt", a single or few words at the end of a paragraph [4].

DG3: Lyric text should be highlighted as it is sung. As human eyes are sensitive to changes in state such as color and motion, applying animated highlighting to the word currently being sung can guide the viewer's eyes to more quickly see and read the right word.

DG4: Lyric text should have sufficient contrast against its background. A sufficient color contrast between the text and its background is important for readability. Alternatively, contrast can also be achieved by styling the text, such as adding outlines, drop shadow, or semi-transparent background boxes.

3.2 Viewer Attention

As the added text requires the viewer's attention to read and comprehend, it is important to coordinate the text with the song audio and video imagery to not split the focus of attention.

DG5: Lyric text should be synchronized to the song. If the lyrics and song are misaligned in timing, the viewer may be distracted by processing similar information more than once. This can cause them to miss important details such as visual cues in the video, negatively affecting their understanding of the content.

DG6: Lyric text should be placed near but not occluding the focus of attention. Prior work has shown that extensive eye movements distract the viewer from understanding details in the video [23] and contribute to eye strain [18]. Thus, text should be placed near the focus of attention to minimize the viewer's eye movement. However, it should also not occlude objects in the focused region as their actions can be crucial to the understanding of the video content. For example, a singer might also be dancing when singing, so the text should avoid occluding any part of the singer's body. Speech bubbles in comics and manga are examples of this idea applied in other media formats.

DG7: Sequential lyric phrases should be placed near each other. Because a lyric phrase is displayed in the video for a limited amount of time, sequential lyric phrases should be placed near each other so that they can be seen without searching. Doing so also minimizes the amount of eye movement [18].

4 AUTOMATED PIPELINE

With the set of design guidelines identified, we have developed a fully automated pipeline that instantiates these guidelines to convert a music video into a lyric video. The input is a video with a song as its background music, such as music videos released by the artists, recordings of live performances, and fan-made lip sync videos and covers. By default, the typeface of the text is Poppins with regular weight, and it is white and 40 pixels in size. The user can adjust the style based on their preferences.

As shown in Figure 3, our pipeline has three stages: text and video contents are preprocessed in stage 1, the spatial placement of the text is computed via an optimization approach in stage 2, and the final lyric video is rendered with animations in stage 3. In stage 1, text and video preprocessing ,we group the song's lyrics into phrases with line breaking (**DG1** and **DG2**) and obtain the in and out timing of each word (**DG5**). We also compute segmentation masks for objects under the focus of attention for every frame of the input video (**DG6**). In stage 2, text placement, we generate the spatial position of each lyric phrase by solving an optimization problem that minimizes energy functions related to saliency (**DG6**), color contrast (**DG4**), and spatial consistency of positions of sequential phrases (**DG7**). Finally, in stage 3, rendering, we render the output lyric video with animated highlighting (**DG3**) based on the spatial and temporal information computed in earlier stages.

4.1 Stage 1: Text and Video Preprocessing

In this stage, lyric phrases are generated by grouping words that are close in time with line breaks added to long text. On the video side, we find and mask out the objects under the attention in every frame. We use the resulting segmentation masks to determine the optimal positions of texts in stage 2.

4.1.1 Generating lyric phrases. We first fetch the song lyrics of the input video from Musixmatch [29]. We then use AutoLyrixAlign [16] to obtain word-level temporal alignment, a pair of time in seconds that specify the in and out times, for each lyric word. In accordance with **DG1**, we group lyric words into individual phrases based on their temporal proximity to each other. More specifically, sequential words are assigned to the same lyric phrase if the difference between the out time of the previous word and the in time of the next word is within a set threshold. By default, the value of this threshold is set to the length of a beat, detected via librosa [27]. Given that there might be inaccuracies in the timing of the words, the user can adjust this threshold. We set the timing of the lyric phrase to start at the in time of its first word and end at the out time of its last word.

4.1.2 Adding line breaks to lyric phrases. To satisfy **DG2**, we add line breaks to lyric phrases with long lines of text. We count the number of characters of every lyric phrase, set the median length as the threshold, and break phrases longer than this threshold into lines that are close in length. Similar to the threshold for grouping phrases, the user can adjust the threshold for line breaking.

4.1.3 Video Shot Detection. As a video is composed of shots and visual contents in a single shot are more consistent, it is easier to extract segmentation masks for objects under attention from each shot separately instead of the entire video. Therefore, we split the input video into successive shots via the approach of Rao et al. [38].

4.1.4 Focus of Attention Segmentation. As human attention is driven in part by "task at hand and current goals" (top-down attention [22]) and drawn to sounding objects [8], we thus look for human figures and objects referenced by the lyrics as objects under the focus of attention. We refer to these objects as foreground objects.

To identify the human figures in the video, we first use an object instance detection model [14] to obtain bounding boxes of people in the first frame of every shot. For every person instance detected, we run a face detection model and extract their facial features represented as a 1D vector [19]. If the cosine similarity of two facial features in two shots is lower than 0.1 (determined empirically), we assign the same person label to these two instances. We then count the number of appearances of each distinct person.

In each shot's first frame, the person with the highest appearance frequency becomes the foreground object, and we input the bounding boxes of their body and face to a video object segmentation model [33] to obtain segmentation masks for their body and face for every frame. In the absence of people or their faces (Figure 4), we look for any object (noun) referenced by the lyric

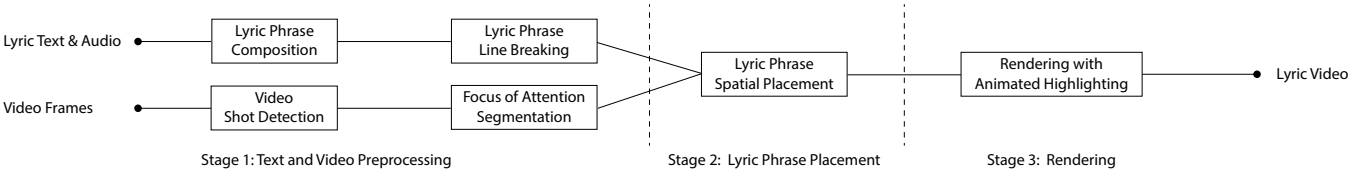

Figure 3: Our pipeline consists of three stages. Given a music video as input, stage 1 aligns the lyrics to the song and groups it into phrases with line breaking. This stage also produces segmentation masks of the objects under the focus of attention. Stage 2 computes the spatial placement of the text via an optimization approach considering terms related to color contrast, attention masks, and the position of the previous phrase. Stage 3 renders the final lyric video with animated highlighting.

Figure 4: For every frame, we generate two segmentation masks. In the absence of a person's body or face, we look for objects referenced by the lyrics (Row 2). We place a lyric phrase near the focus mask while not occluding the foreground mask. *Video source: I Ain't Worried by OneRepublic [32], Ice cream by BLACKPINK & Selena Gomez [5].*

phrases using another object instance detection model pretrained on Objects365 [43]. If no suitable person nor referenced object is detected (e.g. the shot is a b-reel for filler purposes), the output masks are completely black. As shown in Figure 4, we refer to the body segmentation mask as the foreground mask s_{fgd} and the face mask as the focus mask s_{fcs}.

4.2 Stage 2: Lyric Phrase Placement

As shown in Figure 5, in this stage, we find the optimal position p_{min} of a lyric phrase by minimizing a linear combination of energy functions implemented according to the design guidelines.

To compute values for the terms in our total energy function, we first collect the set of frames $\mathcal{F}$ spanned by the in and out times of a lyric phrase. For the frames in $\mathcal{F}$, we compute the pixel-wise average of their focus mask s_{fcs} and foreground mask s_{fgd} (Figure 4) to obtain an average focus mask $\bar{s}_{\text{fcs}}$ and an average foreground mask $\bar{s}_{\text{fgd}}$ (Row 2 of Figure 5). Furthermore, we obtain the background b of a frame f by inverting the foreground mask and multiplying by f so that $b = f \cdot (1 - s_{\text{fgd}})$. We then compute the pixel-wise average of all backgrounds of frames in $\mathcal{F}$ to obtain an average background image $\bar{b}$. We compute averages because we optimize for a fixed text position over the time span of a lyric phrase. Finally, we rasterize the text of a lyric phrase into a greyscale image k, stored as a 2D array of floats ranging from 0 to 1. We use k as a convolution kernel in the energy terms. We describe specific energy terms below, with p denoting a candidate pixel coordinate for placing the lyric phrase (upper left corner of the text rectangle).

Placement near focus of attention. The first two energy functions in our optimization, E_{fcs} and E_{fgd}, correspond to **DG6**. E_{fcs} puts the lyric phrase position p close to the visual center of mass p_{center} of $\bar{s}_{\text{fcs}}$, which is the average position of the pixel intensities in $\bar{s}_{\text{fcs}}$:

$$E_{\text{fcs}}(p) = ||p_{\text{center}} - p||_2$$

E_{fgd} ensures that the lyric text minimally occludes the white regions in the average foreground mask $\bar{s}_{\text{fgd}}$. To achieve this, we convolve the text kernel k with $\bar{s}_{\text{fgd}}$ to produce a foreground overlap cost map $o = \bar{s}_{\text{fgd}} \otimes k$. Since entries in k have values from 0 to 1, $o(p)$ returns the weighted sum of the pixels in $\bar{s}_{\text{fgd}}$ that overlaps with the text if it is placed at p. Therefore, the value of $o(p)$ is lower when there is less overlap (Figure 5). Our energy function E_{fgd} is then computed as:

$$E_{\text{fgd}} = o(p)$$

Placement with high color contrast. E_{cnt} places the text against background regions with high color contrast (**DG4**). Similar to E_{fgd}, we compute a background color difference cost map $c(p)$ that returns the color difference value between the text and its background if it is placed at position p. To obtain c, we first compute a background color difference image $\bar{b}_{\text{diff}}$ by subtracting the text color from the average background $\bar{b}$ (Row 2 of Figure 5). The difference between two colors is defined as the Euclidean distance between two RGB colors. We then convolve the color difference map with the inverted text kernel $1 - k$. We invert the kernel so that the color differences of pixels surrounding the text, instead of underneath the text, are taken into consideration. The resulting $c' = \bar{b}_{\text{diff}} \otimes (1 - k)$ has entries with values equal to the sum of the color differences surrounding the text. We then invert c' by subtracting its maximum value from it to obtain c so that lower value means higher color contrast. Our energy function E_{cnt} is thus:

$$E_{\text{cnt}} = c(p)$$

Placement near previous lyric phrase. The energy function E_{prv} is designed to place each lyric phrase near the previous one (**DG7**). Given the position of the previous phrase p_{prev}, E_{prv} is defined as:

$$E_{\text{prv}} = ||p_{\text{prev}} - p||_2$$

Note that, for the very first lyric phrase and the first lyric phrase in a shot, we set $E_{\text{prv}} = 0$. We do not let the previous phrase influence

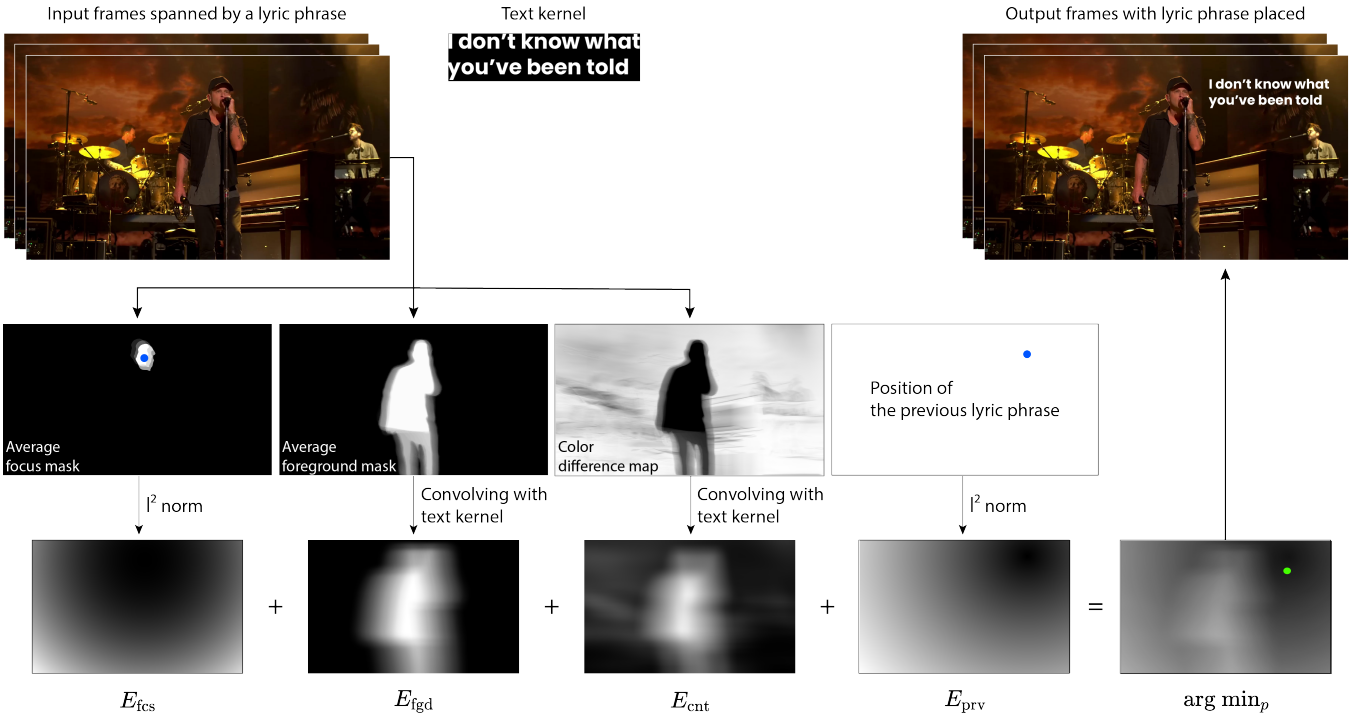

Figure 5: *Visualization of the lyric phrase placement algorithm.* Given a lyric phrase and its associated set of video frames, the placement algorithm minimizes a linear combination of energy functions related to the design guidelines to find the optimal position for that lyric phrase. Note that darker pixels correspond to lower scores, and the cost maps associated with the energy functions are smaller than the input frame size because we do not allow the text to be placed partially or fully out of the screen. *Video source: I Ain't Worried by OneRepublic [32].*

the positioning of the current one because the composition of video imagery often changes significantly from shot to shot.

Together, we combine the individual energy functions described above to define the following optimization function:

$$p_{\text{min}} = \arg\min_p \left(w_{\text{fcs}} E_{\text{fcs}} + w_{\text{fgd}} E_{\text{fgd}} + w_{\text{cnt}} E_{\text{cnt}} + w_{\text{prv}} E_{\text{prv}} \right)$$

where each weight w is adjustable for fine-tuning the positioning.

4.3 Stage 3: Rendering

At the end of stage 2, every lyric phrase has a coordinate p for its placement in the video and in and out times for the whole phrase and each individual word. We write a script in Adobe ExtendScript to automatically parse and load the lyric phrase data into After Effects and overlay them with animated highlighting (**DG3**) onto the original music video. To further support **DG6**, we use the foreground segmentation masks to overlay the foreground objects on top of the text. The user can choose not to apply these features.

We provide a set of commonly used highlighting animations (**DG3**) designed based on existing lyric videos. All text in a phrase can be animated in and out together, either through fading, sliding up or down, or a combination of both. We add extra padding time (0.2 seconds, adjustable) to the in and out times of a phrase for the animations to take place. Doing so also gives the viewer extra time to read the last few words in a phrase if they are sung shortly. Individual words can be highlighted via fading in and out, sliding up and down, changing to an accent color, or a combination of them. The default highlighting is that a phrase fades in and out with individual words sliding up when sung. We apply these animations by automatically inserting keyframes at appropriate times via ExtendScript. Rendering the video in After Effects additionally allows the user to edit it further by taking advantage of the comprehensive set of tools that After Effects provides.

4.4 Example Usage Scenario

To obtain a lyric video, the user first inputs a music video into the pipeline for a result. They can then adjust the default settings by editing values in a JSON file. For example, the user can adjust the text font, color, and animation, or increase weights such as w_{fcs} and w_{fgd} to strongly draw the text to the focus of attention (**DG6**). They can also fix pipeline errors as described in Section A. After the desired adjustments are made, they can rerun the pipeline for a new result. They can repeat this process multiple times, or make one-off or more detailed adjustments in After Effects.

5 RESULTS

To demonstrate the efficacy of our automated pipeline, we have generated 15 lyric videos from music videos on YouTube. Please find the resulting videos at https://hci.stanford.edu/research/lyricvideo/.

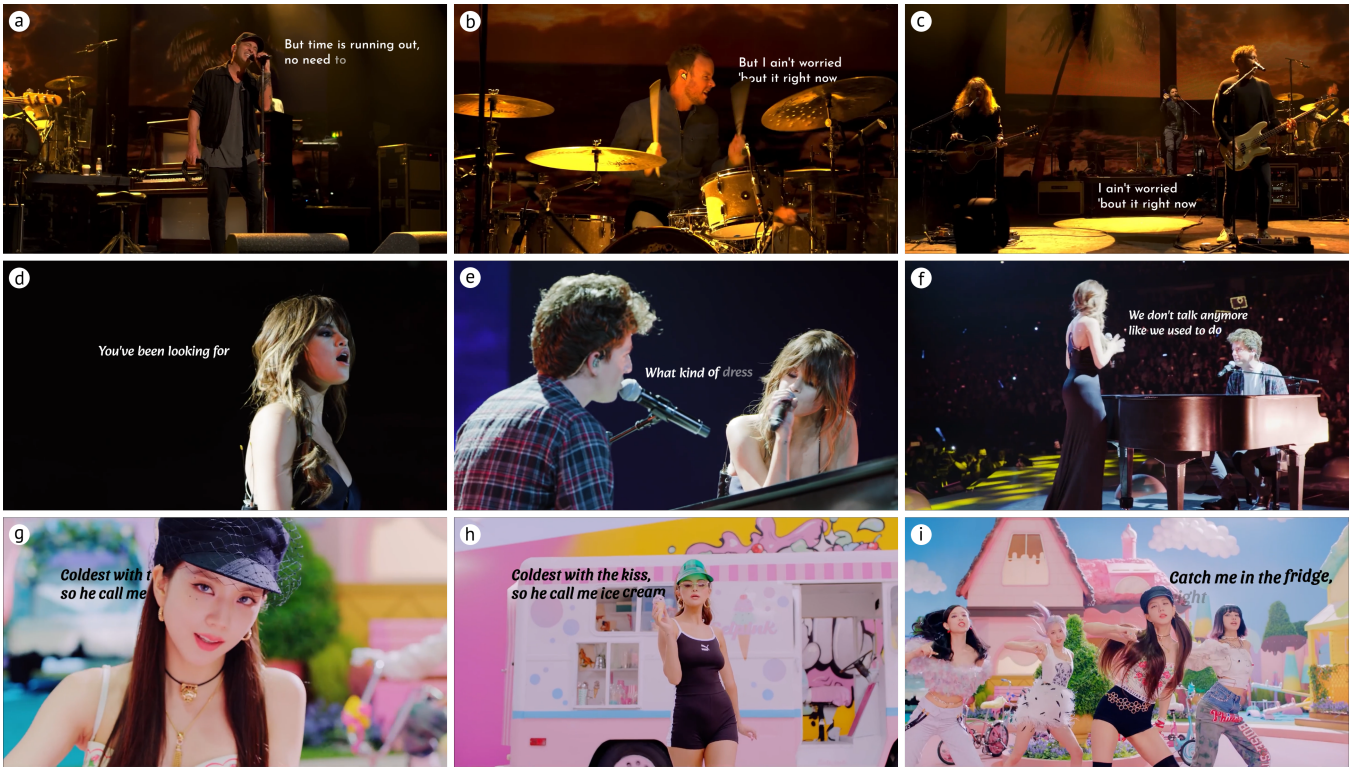

Figure 6: We present 3 lyric videos automatically generated by our pipeline from inputs that are challenging to add text to. In the first video (a-c), the camera constantly switch among the musicians while also zooming in and out. In the second video (d-f), two singers are present and the main female singer constantly walks around the stage. The third video (g-i) features many quick shot changes with significant differences in composition, as well as many musicians. In these cases, our pipeline is able to generate results that adhere closely to our design guidelines. ***Video sources (from top to bottom): I Ain't Worried by OneRepublic [32], We Don't Talk Anymore by Charlie Puth & Selena Gomez [37], Ice Cream by BLACKPINK & Selena Gomez [5].***

Figure 1 and Figure 6 feature 9 of the resulting videos. We select a variety of input videos, including official music videos, live performances (Figure 1c, Figure 6a-f), and animated features (Figure 1d). The songs are from diverse genres with tempos ranging from slow to fast. The arrangement of the musicians varies from one singer, a singer with a band, to multiple singers (Figure 6). We also use videos in both horizontal and vertical aspect ratios (Figure 1c and f). All the results presented are generated by our fully automated pipeline without any manual edits in After Effects (some input parameters, like text style and animation, are adjusted for certain examples) .

In all these examples, our pipeline consistently finds and places text next to the viewer's focus of attention (E_{fcs} for **DG6**), such as the ice cream cone held by the singer in Figure 6h. In the case of Figure 1a where multiple faces are present, our pipeline also correctly identifies the main singer's face via appearance frequency counting. Our pipeline is also able to identify spare regions, an area with minimal foreground actions (E_{fgd} for **DG6**) and good color contrast (E_{cnt} for **DG4**) in the video. In Figure 1b, our pipeline places the text in the dark-colored window right next to the singer. In Figure 1d, as the singer swings her arm from the lower left to top right, the pipeline finds the spare region in between her face and her arms for the text. Similarly, the text sits at a nice dark area near the center of the video frame Figure 6c.

In addition to horizontal aspect ratio videos, our pipeline can also add text to vertical videos that are popular on social media platforms like TikTok and Instagram. Figure 1c and f are two such examples. By breaking a lyric phrase into multiple lines, our pipeline is able to fit the text into the narrow width of the video and places it near the singer with good contrast.

Figure 6 presents three examples whose input music videos are challenging to process. In the first video of a live performance (Figure 6a-c), the singer is accompanied by a band of musicians, and the video camera switches back and forth to feature different people while also zooming in and out. Our pipeline can consistently identify the other musicians in the absence of the singer (**DG6**) and avoid bright-colored regions (**DG4**) for placing white text (Figure 6b-c). In the second video (Figure 6d-f), the main female singer walks around the male singer playing the piano, and the camera occasionally switches to showing the concert audience. Our pipeline places the text to consistently follow the female singer and finds space for the lyric text in between the two singers when they are close together (Figure 6e-f).

The third video (Figure 6g-i) is even more challenging than the previous two in that it is a fast-tempo song featuring 5 singers with dynamic dance movements. Shot changes occur very frequently, as one lyric phrase often spans three or more shots. Figure 6g-h shows an example of a lyric phrase spanning multiple shots. In Figure 6g, the lyric phrase is briefly, partially blocked by the foreground object. Such occlusion ensures that the text does not distract the viewer from the focus of attention, while the impact on readability is minimal since the word currently being sung ("me") is still visible. After just a few frames, the shot quickly changes to the one shown in Figure 6h. Even though the singer's position changes and the camera zooms out, the text remains near the focus of attention.

Overall, given a wide variety of input videos, our pipeline is able to produce lyric videos that closely follow the design guidelines and makes the text look like it is an integral part of the video.

6 EVALUATION

We conducted a user evaluation to investigate the following research question: How well does the lyric video produced by our automated pipeline achieve the two overarching goals of our proposed design guidelines, ensuring text readability and maintaining unified focus of attention?

6.1 Materials

To examine the research question, for a given input music video, we generated 4 conditions: a *Baseline* condition in the style of video subtitles, a *Full* condition generated by our automated pipeline, and two conditions, *Readability Ablated* and *Attention Ablated*, generated by pipelines in which we ablated certain components based on relevant design guidelines.

The Baseline condition is made to look like video subtitles. Each lyric phrase corresponds to one line in the original lyrics file retrieved from Musixmatch with no line breaks and always sits at the bottom center of the video. The Readability Ablated condition does not incorporate design guidelines related to text readability (**DG1**-**DG4**); in this version, a lyric phrase is composed in the same way as the Baseline condition. We do not add animated highlighting and do not include the color contrast energy function (i.e., we set $w_{\text{cnt}} = 0$) in spatial placement. The Attention Ablated condition does not consider design guidelines on unifying attention (**DG6**-**DG7**). We still sync the lyric phrases to the song, but we do not place them near the focus of attention (i.e., $w_{\text{fcs}} = w_{\text{fgd}} = 0$) and sequential lyric phrases are not placed near each other (i.e., $w_{\text{prv}} = 0$).

We chose 10 of the input music videos from Section 5 for the evaluation (full video list in Section B) and created a total of 40 videos. The generated videos range from 30 to 60 seconds in length (only sections of the original videos are used to control the total duration of the survey).

6.2 Procedure

We distributed our online survey via multiple listservs for a wide range of participants from the US. Each survey presented a participant with the 4 conditions of one video randomly chosen from the 10 videos in our evaluation set. The viewing sequence of the 4 conditions was randomized to mitigate the learning effect. After viewing each video, we asked the participant to rate the following statements on a 7-point Likert scale and elaborate on their choices in a free response form.

Q1: The text of the lyrics in the video is easy to read.
Q2: I can both easily read the text and watch the video imagery.
Q3: The overall viewing experience is good.

The scale ranged from Strongly Disagree (1) to Strongly Agree (7) for all questions. At the end of the survey, we also collected the participant's age and gender. The participants did not receive any monetary compensation for completing the survey.

6.3 Evaluation Results

We received 57 valid responses to the online survey after removing duplicated answers. The participants (27 female, 28 male, 1 non-binary, 1 prefer not to say) range from 19 to 32 years old ($\bar{x} = 25.35, \text{SD} = 2.03$). Each of the 10 videos was shown to at least 5 participants. We aggregate ratings on the same Likert scale question for the same condition of a video and show the results in Figure 7.

For each of the three questions (Q1-3), we first conducted a Friedman's Test which shows that there is significant difference among the 4 conditions ($p < 0.001$ for all questions). We then ran post-hoc pairwise two-tailed Wilcoxon signed-rank tests with Holm-Bonferroni correction to compare the Full condition against the other three. The full Wilcoxon statistics can be found in Table 1 in the appendix. For Q1, the Full condition ($\bar{x} = 6.80$) is rated significantly higher than the other three: vs. Baseline ($\bar{x} = 6.28$): $p = 0.043$; vs. Readability Ablated ($\bar{x} = 5.89$): $p = 0.0004$; vs. Attention Ablated ($\bar{x} = 6.14$): $p = 0.0003$. For Q2, the Full condition ($\bar{x} = 6.54$) performances significantly better than the other three: vs. Baseline ($\bar{x} = 6.05$): $p = 0.034$; vs. Readability Ablated ($\bar{x} = 5.70$): $p = 0.001$; vs. Attention Ablated ($\bar{x} = 5.47$): $p = 0.000006$. For Q3, the Full condition ($\bar{x} = 6.45$) is rated significantly higher than the two ablated conditions: vs. Readability Ablated ($\bar{x} = 5.64$): $p = 0.003$; vs. Attention Ablated ($\bar{x} = 5.70$): $p = 0.0006$, and is not significantly different from the Baseline ($\bar{x} = 6.17$).

Overall, these results indicate that the Full condition produced by our pipeline is significantly better than the ablated conditions in terms of maintaining readability, unifying attention, and overall experience (Figure 7). Moreover, the Full condition is also significantly better than the Baseline condition in the first two measures.

The participant comments provide insights into the quantitative results. For the Full version, without knowing the design guidelines, 14 participants specifically commented that the placement of the text is close to their focus of attention: "The text is very close to where my attention of the video would be" (P22). This helped them to both easily read the text ("Text was easy to follow", P19) and pay attention to the text and video together ("It was easy to saccade back and forth", P25). P57 mentioned that placing sequential phrases in the same shot near each other helps with readability: "Easier to find this time as some of them start in the same place."

In the Readability Ablated version, 11 participants found that the text is still near their focus of attention, but certain phrases are too long to be easily read and overlap with objects under their attention: "the lyrics will occlude the main subject and each segment is too long" (P38). P29 additionally found that the lack of highlighting animation negatively affects the text readability: "Locating the text was also harder because there's no movement."

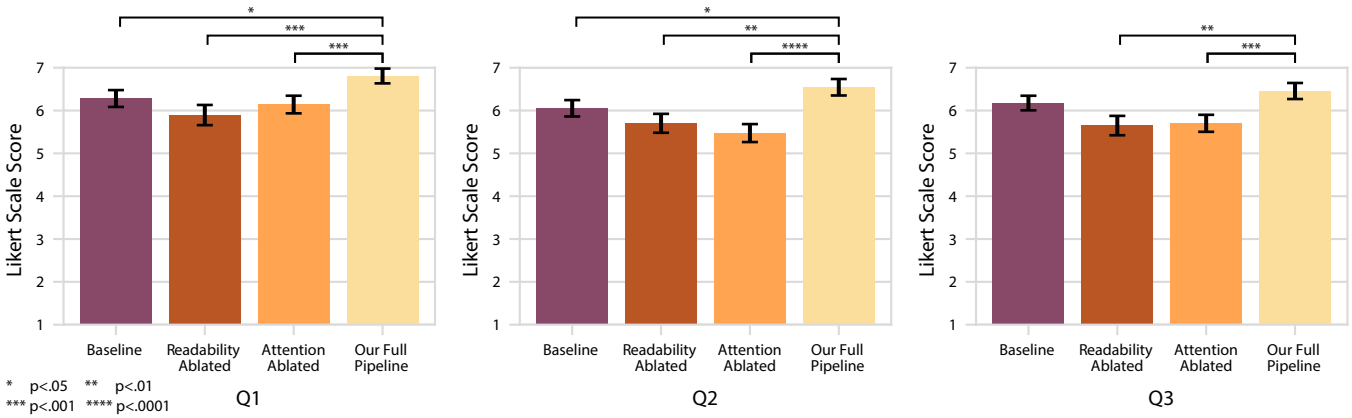

Figure 7: ***Aggregated Likert scale ratings of questions on text readability (Q1), unified attention (Q2), and overall experience (Q3).*** **Horizontal brackets indicate pairwise significant difference, and the error bars show standard error. Overall, the results demonstrate that the lyric videos produced by our pipeline are significantly better at achieving the goals of ensuring lyrics readability and unifying the viewer's focus of attention than the other versions.**

The main issue with the Attention Ablated condition is that lyric phrases are placed often far away from the focus of attention, making them hard to find. 11 participants mentioned this concern: "Can't really focus on the video because I have to spend time looking for the text " (P46), and "Can only focus on either one" (P42). On a positive note, 4 participants did find the animated highlighting, which is also applied in the Full version, helpful for readability: "The way each word was emphasized along with the singing freed up some processing load" (P44).

The participants rated that the Baseline condition provides an overall good viewing experience similar to the Full condition. Familiarity with the subtitle text contributes to the Baseline condition's high rating, as 12 participants mentioned that they knew the text would appear at a fixed location: "This pattern is very comfortable and common" (P57). However, some also noted that this trades creative expression for predictability (P53: "this feels like subtitles, not like a lyric video"). Despite the similar ratings, the participants used more positive words for the Full condition, such as "enjoy", "easy", and "fun", and instead described Baseline as "normal", "standard", and "familiar" (P19: "not an outstanding viewing experience"). Moreover, the Baseline condition also shares the same issue as the Attention Ablated version. 8 participants stated that their eyes need to move around a lot in order to read the text and watch the video: "Sometimes I need to skip some lines to focus on the singer" (P10) and "Eyes have to move a lot to look at other places" (P3).

Overall, significant differences in the Likert scale ratings and the participant comments demonstrate that the lyric video generated by our automated pipeline, which instantiates our design guidelines, are effective in achieving our two overall goals, ensuring text readability and unifying the viewer's focus of attention.

7 LIMITATIONS AND FUTURE WORK

We acknowledge that the 10 input videos in our user evaluation forms a relatively small corpus, and there is a lack of comparison to user-made lyric videos.

We discuss common failure cases of our pipeline in detail in Section A. One area that future work can focus on is improving our focus of attention segmentation algorithm. Future work can explore integrating deep learning-based saliency detection models that use bottom-up attention cues, such as motion and color, into our current top-down approach. Moreover, a multi-modal approach can be considered, such as identify sounding instruments or the singer of the current phrase in case of multiple singers.

Our pipeline works best when there are enough spare regions (places with minimal foreground actions and sufficient color contrast) in the input video (more discussed in Section A). Some stylized videos are purposefully designed this way, and many vertical videos also do not have such regions since a human face or body can often take up the majority of the screen. Future work might explore means to modify existing video content to make space for lyric text, such as finding and blurring the region in a frame with the least amount of visual information.

8 CONCLUSION

Lyric videos are widely produced today despite the amount of time and careful coordinations they require to make. We identify 7 design guidelines to help creators ensure that the text in these videos are readable and the viewer's focus of attention are unified. We further implement a fully automated pipeline that converts an input music video into a lyric video following these design guidelines. We demonstrate the efficacy of our pipeline by generating lyric videos from music videos that vary significantly in format and imagery. A 57-respondent user study shows that lyric videos produced by our pipeline are effective in achieving our goals of ensuring text readability and maintaining unified focus of attention.

ACKNOWLEDGMENTS

We would like to thank John Nelson for his inputs during the early stage of this project. This work was supported by the Stanford Graduate Fellowship and Brown Institute for Media Innovation.

REFERENCES

[1] Maneesh Agrawala, Wilmot Li, and Floraine Berthouzoz. 2011. Design Principles for Visual Communication. *Commun. ACM* 54, 4 (apr 2011), 60–69. https://doi.org/10.1145/1924421.1924439

[2] Aitor Álvarez, Haritz Arzelus, and Thierry Etchegoyhen. 2014. Towards Customized Automatic Segmentation of Subtitles. In *Advances in Speech and Language Technologies for Iberian Languages*. Springer, Cham, 229–238.

[3] Sudan Archives. 2022. Sudan Archives - Selfish Soul (Official Video). Retrieved July 24, 2023 from https://youtu.be/eaY8kI0oEpA

[4] David R Bennett. 2002. *Meant to be read: Typesetting principles for the digital age*. Lamar University-Beaumont, Beaumont, Texas.

[5] BLACKPINK and Selena Gomez. 2021. BLACKPINK - 'Ice Cream (with Selena Gomez)' M/V. Retrieved July 24, 2023 from https://youtu.be/vRXZj0DzXIA

[6] Andy Brown, Rhia Jones, Mike Crabb, James Sandford, Matthew Brooks, Mike Armstrong, and Caroline Jay. 2015. Dynamic Subtitles: The User Experience. In *Proceedings of the ACM International Conference on Interactive Experiences for TV and Online Video (TVX '15)*. Association for Computing Machinery, New York, NY, USA, 103–112. https://doi.org/10.1145/2745197.2745204

[7] The Chainsmokers. 2022. The Chainsmokers - iPad (Live at SUMMIT at One Vanderbilt). Retrieved July 24, 2023 from https://youtu.be/w1DZWaOHImk

[8] Chenglizhao Chen, Mengke Song, Wenfeng Song, Li Guo, and Muwei Jian. 2023. A Comprehensive Survey on Video Saliency Detection With Auditory Information: The Audio-Visual Consistency Perceptual is the Key! *IEEE Transactions on Circuits and Systems for Video Technology* 33, 2 (2023), 457–477. https://doi.org/10.1109/TCSVT.2022.3203421

[9] Ho Kei Cheng and Alexander G. Schwing. 2022. XMem: Long-Term Video Object Segmentation with an Atkinson-Shiffrin Memory Model. In *ECCV 2022*, Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (Eds.). Springer, Cham, 640–658.

[10] Coldplay. 2022. Coldplay - The Scientist. Retrieved July 24, 2023 from https://www.youtube.com/shorts/KsyALRZSn2o

[11] Miley Cyrus. 2023. Miley Cyrus - Flowers (Official Video). Retrieved July 24, 2023 from https://youtu.be/G7KNmW9a75Y

[12] The Described and Captioned Media Program. 2023. Guidelines and Best Practices for Captioning Educational Video. https://dcmp.org/learn/captioningkey

[13] Bob Dylan. 1965. Bob Dylan - Subterranean Homesick Blues (Official HD Video). https://youtu.be/MGxjIBEZvx0

[14] C. Feng, Y. Zhong, Y. Gao, M. R. Scott, and W. Huang. 2021. TOOD: Task-aligned One-stage Object Detection. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE Computer Society, Los Alamitos, CA, USA, 3490–3499. https://doi.org/10.1109/ICCV48922.2021.00349

[15] Olivia Gerber-Morón and Agnieszka Szarkowska. 2018. Line breaks in subtitling: an eye tracking study on viewer preferences. *Journal of eye movement research* 11, 3 (2018), 18 pages.

[16] Chitralekha Gupta, Emre Yılmaz, and Haizhou Li. 2020. Automatic Lyrics Alignment and Transcription in Polyphonic Music: Does Background Music Help?. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, Barcelona, Spain, 496–500. https://doi.org/10.1109/ICASSP40776.2020.9054567

[17] Srinidhi Hegde, Jitender Maurya, Ramya Hebbalaguppe, and Aniruddha Kalkar. 2020. SmartOverlays: A Visual Saliency Driven Label Placement for Intelligent Human-Computer Interfaces. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, Snowmass, CO, USA, 1110–1119. https://doi.org/10.1109/WACV45572.2020.9093587

[18] Yongtao Hu, Jan Kautz, Yizhou Yu, and Wenping Wang. 2015. Speaker-Following Video Subtitles. *ACM Trans. Multimedia Comput. Commun. Appl.* 11, 2, Article 32 (jan 2015), 17 pages. https://doi.org/10.1145/2632111

[19] Q. Huang, Y. Xiong, and D. Lin. 2018. Unifying Identification and Context Learning for Person Recognition. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Los Alamitos, CA, USA, 2217–2225. https://doi.org/10.1109/CVPR.2018.00236

[20] Adobe Inc. 2023. Premiere Pro. https://www.adobe.com/products/premiere.html

[21] Jun Kato, Tomoyasu Nakano, and Masataka Goto. 2015. TextAlive: Integrated Design Environment for Kinetic Typography. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. Association for Computing Machinery, New York, NY, USA, 3403–3412. https://doi.org/10.1145/2702123.2702140

[22] Fumi Katsuki and Christos Constantinidis. 2014. Bottom-Up and Top-Down Attention: Different Processes and Overlapping Neural Systems. *The Neuroscientist* 20, 5 (2014), 509–521. https://doi.org/10.1177/1073858413514136 arXiv:https://doi.org/10.1177/1073858413514136 PMID: 24362813.

[23] Kuno Kurzhals, Emine Cetinkaya, Yongtao Hu, Wenping Wang, and Daniel Weiskopf. 2017. Close to the Action: Eye-Tracking Evaluation of Speaker-Following Subtitles. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. Association for Computing Machinery, New York, NY, USA, 6559–6568. https://doi.org/10.1145/3025453.3025772

[24] Kuno Kurzhals, Fabian Göbel, Katrin Angerbauer, Michael Sedlmair, and Martin Raubal. 2020. *A View on the Viewer: Gaze-Adaptive Captions for Videos*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3313831.3376266

[25] John Legend. 2022. John Legend - Nervous (Live in Las Vegas). Retrieved July 24, 2023 from https://youtu.be/V5vLVPQ-S-0

[26] Google LLC. 2023. YouTube Studio. https://studio.youtube.com/

[27] Brian McFee, Colin Raffel, Dawen Liang, Daniel Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. 2015. librosa: Audio and Music Signal Analysis in Python. In *Proceedings of the 14th Python in Science Conference*. Scipy, USA, 18–24. https://doi.org/10.25080/majora-7b98e3ed-003

[28] Idina Menzel. 2013. Idina Menzel - Let It Go (from Frozen) (Official Video). Retrieved July 24, 2023 from https://youtu.be/YVVTZgwYwVo

[29] Musixmatch. 2023. Musixmatch. https://www.musixmatch.com/

[30] Netflix. 2023. English Timed Text Style Guide. https://partnerhelp.netflixstudios.com/hc/en-us/articles/217350977-English-Timed-Text-Style-Guide

[31] NFL. 2023. Rihanna's FULL Apple Music Super Bowl LVII Halftime Show. Retrieved July 24, 2023 from https://youtu.be/HjBo--1n8lI

[32] OneRepublic. 2022. OneRepublic - I Ain't Worried (Live From The Tonight Show Starring Jimmy Fallon). Retrieved July 24, 2023 from https://youtu.be/fDuNemLHGzw

[33] Matthieu Paul, Martin Danelljan, Christoph Mayer, and Luc Van Gool. 2022. Robust Visual Tracking By Segmentation. In *European Conference on Computer Vision (ECCV)*. Springer-Verlag, Berlin, Heidelberg, 571–588. https://doi.org/10.1007/978-3-031-20047-2_33

[34] Elisa Perego. 2008. What Would We Read Best? Hypotheses and Suggestions for the Location of Line Breaks in Film Subtitles. *The Sign Language Translator and Interpreter* 2 (2008), 35–63.

[35] Katy Perry. 2013. Katy Perry - Birthday. https://youtu.be/jqYxyd1iSNk

[36] Prince. 1987. Prince - Sign O' The Times. https://youtu.be/8EdxM72EZ94

[37] Charlie Puth and Selena Gomez. 2016. Charlie Puth & Selena Gomez - We Don't Talk Anymore [Official Live Performance]. Retrieved July 24, 2023 from https://youtu.be/i_yLpCLMaKk

[38] A. Rao, L. Xu, Y. Xiong, G. Xu, Q. Huang, B. Zhou, and D. Lin. 2020. A Local-to-Global Approach to Multi-Modal Movie Scene Segmentation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Los Alamitos, CA, 10143–10152. https://doi.org/10.1109/CVPR42600.2020.01016

[39] Keith Rayner. 1975. The perceptual span and peripheral cues in reading. *Cognitive psychology* 7, 1 (1975), 65–81.

[40] Reddit. 2023. r/HighQualityGifs. http://reddit.com/r/HQGStudios

[41] Rihanna. 2009. Rihanna - Umbrella (Orange Version) (Official Music Video) ft. JAY-Z. Retrieved July 24, 2023 from https://youtu.be/CvBfHwUxHIk

[42] Rihanna. 2010. Rihanna - Only Girl (In The World). Retrieved July 24, 2023 from https://youtu.be/pa14VNsdSYM

[43] S. Shao, Z. Li, T. Zhang, C. Peng, G. Yu, X. Zhang, J. Li, and J. Sun. 2019. Objects365: A Large-Scale, High-Quality Dataset for Object Detection. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE Computer Society, Los Alamitos, CA, USA, 8429–8438. https://doi.org/10.1109/ICCV.2019.00852

[44] Ed Sheeran. 2022. Ed Sheeran - Sandman. Retrieved July 24, 2023 from https://youtube.com/shorts/0T5yt0MzmdQ

[45] Taylor Swift. 2021. Taylor Swift - august (studio sessions). Retrieved July 24, 2023 from https://youtu.be/pc_2ZKB4LVc

[46] Taylor Swift. 2022. Taylor Swift - Anti-Hero. https://youtu.be/XqN2qFvY64U

[47] Tan Tang, Junxiu Tang, Jiewen Lai, Lu Ying, Yingcai Wu, Lingyun Yu, and Peiran Ren. 2022. SmartShots: An Optimization Approach for Generating Videos with Data Visualizations Embedded. *ACM Trans. Interact. Intell. Syst.* 12, 1, Article 4 (mar 2022), 21 pages. https://doi.org/10.1145/3484506

[48] Friederike Tegge and Katharina Parry. 2020. The impact of differences in text segmentation on the automated quantitative evaluation of song-lyrics. *Plos one* 15, 11 (2020), e0241979.

[49] Quoc V. Vy, Jorge A. Mori, David W. Fourney, and Deborah I. Fels. 2008. EnACT: A Software Tool for Creating Animated Text Captions. In *Computers Helping People with Special Needs*, Klaus Miesenberger, Joachim Klaus, Wolfgang Zagler, and Arthur Karshmer (Eds.). Springernet, Berlin, Heidelberg, 609–616.

[50] Fangzhou Wang, Hidehisa Nagano, Kunio Kashino, and Takeo Igarashi. 2017. Visualizing Video Sounds With Sound Word Animation to Enrich User Experience. *IEEE Transactions on Multimedia* 19, 2 (2017), 418–429. https://doi.org/10.1109/TMM.2016.2613641

[51] Waxahatchee. 2020. Waxahatchee - Fire (Official Video). Retrieved July 24, 2023 from https://youtu.be/cEyYlyRr2_U

[52] Waxahatchee. 2020. Waxahatchee - Lilacs (Official Video). Retrieved July 24, 2023 from https://youtu.be/OaA7I7B1pOk

[53] Gareth Ford Williams. 2009. BBC Online Subtitling Editorial Guidelines. https://www.bbc.co.uk/accessibility/forproducts/guides/subtitles

[54] YouTube. 2023. HQG Studios. https://www.youtube.com/@HQGStudios

[55] Sean Zdenek. 2015. *Reading Sounds: Closed-Captioned Media and Popular Culture*. University of Chicago Press, Chicago, IL, USA.

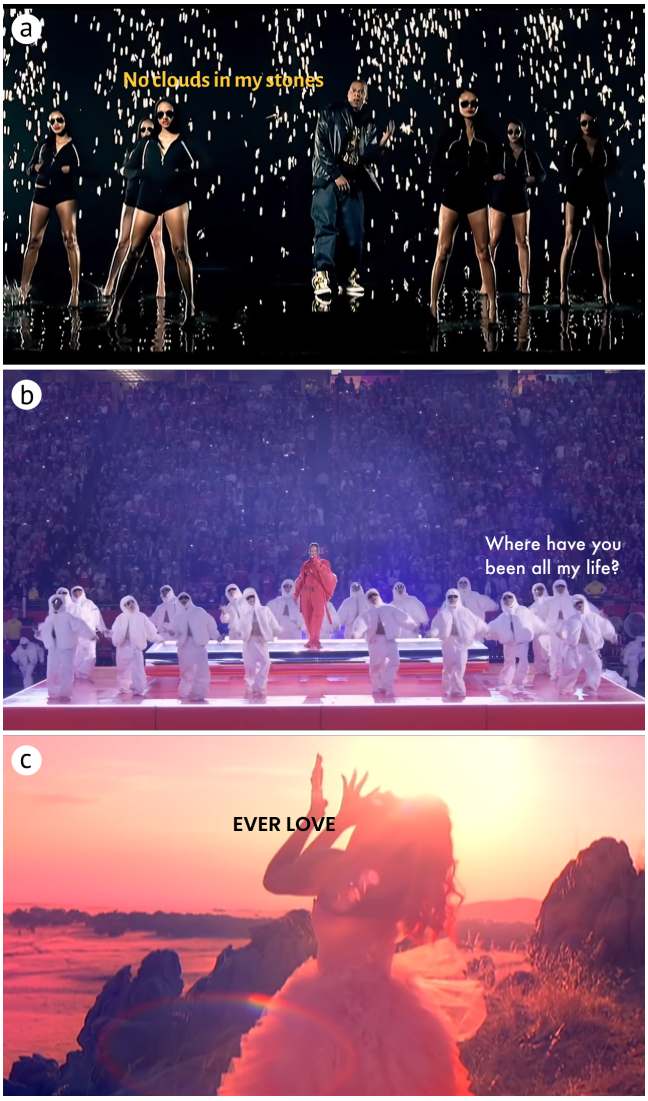

Figure 8: We find three recurring failure cases when generating lyric videos with our pipeline. When the input video is visually cluttered, it is hard to place the text in an easy-to-read way (a). Our focus of attention algorithm might find the wrong focal person when many similarly looking people are present (b). Our video object segmentation model [24] does not work well when the foreground and background are not clearly separated (c). *Video sources: (a) Umbrella by Rihanna ©2007 The Island Def Jam Music Group [41], (b) Rihanna's FULL Apple Music Super Bowl LVII Halftime Show [31], (c) Only Girl (In The World) by Rihanna ©2010 The Island Def Jam Music Group [42].*

A FAILURE CASES

We tested many videos with our automated pipeline and found 3 recurring failure cases (Figure 8).

Case 1: High visual clutter If the input video is visually cluttered, our pipeline might not place the text in an easy-to-read way because it is difficult to ensure good contrast. This is shown in Figure 8a, where bright raindrops appear against a dark background. As discussed in Section 7, one possible fix in the future is to make a region of the background blurred or in solid color.

Case 2: Focus of attention algorithm limitations Our algorithm looks for people or objects referenced by the lyrics. When they cannot be detected, the algorithm outputs empty masks that can result in text placed far from the focus of attention (Section 4.1.4). Further, we define the person who appeared most frequently as the foreground object. We may find the wrong focal person when the correct one is surrounded by many similarly looking people, like the dancers in Figure 8b. Currently, the user can fix these by drawing a bounding box around the desired foreground object in a shot's first frame.

Case 3: Tracking model limitations As discussed in Section 7, our video object segmentation model [33] does not work well on some videos. The video shown in Figure 8c has a pink tone that mixes the foreground and background, while the foreground object also has fast motions. The model cannot consistently track the foreground object. This results in text placements that occlude the foreground object. Currently, this can be manually fixed by tracking models that supports interactive editing [9].

B VIDEOS USED IN USER EVALUATION

(1) I Ain't Worried by OneRepublic [32]
(2) We Don't Talk Anymore by Charlie Puth & Selena Gomez [37]
(3) Ice Cream by BLACKPINK & Selena Gomez [5]
(4) Selfish Soul by Sudan Archives [3]
(5) August by Taylor Swift [45]
(6) Nervous by John Legend [25]
(7) Lilacs by Waxahatchee [52]
(8) Fire by Waxahatchee [51]
(9) Flower by Miley Cyrus [11]
(10) Let It Go by Idina Menzel [28]

C USER EVALUATION STATISTICS

Q1	Baseline	Readability Ablated	Attention Ablated
Full Pipeline	z = 277.5 p = 0.043	z = 121.5 p = 0.0004	z = 88.5 p = 0.0003

Q2	Baseline	Readability Ablated	Attention Ablated
Full Pipeline	z = 254.5 p = 0.034	z = 222 p = 0.001	z = 96 p = 0.000006

Q3	Baseline	Readability Ablated	Attention Ablated
Full Pipeline	z = 309 p = 0.164	z = 235 p = 0.003	z = 150 p = 0.0006

Table 1: Statistics of the post-hoc pairwise Wilocoxon test.