November 23, 2016

REMOVING UNDESIRABLE AUDIO FEATURES FROM A MEDIA STREAM

Luke Macpherson

REMOVING UNDESIRABLE AUDIO FEATURES FROM A MEDIA STREAM

ABSTRACT

A feature correction system is used to remove undesirable audio and video features from a media stream. The system determines an undesirable audio feature in the media stream by detecting a region that is relatively silent or has a predetermined speech disfluency, e.g., umms, y'knows, or ahhs. Further, the system establishes a start time and an end time for the undesirable audio feature. The system then detects video frames in the media stream that include the undesirable audio feature. The system finally performs feature correction for the undesirable audio feature and/or the video frames according to one or more predetermined techniques, e.g., muting the undesirable audio feature and/or compressing related video frames.

PROBLEM STATEMENT

There is typically a large disparity between the rate at which humans can generate meaningful spoken content and the rate at which they can consume spoken content. Frequently in recorded speech, this disparity manifests itself as frustration, where a listener must tolerate the long silences, umms, throat clearings, y'knows, and ahhs generated by a speaker. For example, a user watching a recorded amateur video given by a casual speaker often listens or waits while a presenter takes long pauses to think about what they want to say next. In a present scenario, a typical workflow for removing the undesirable regions of speech include loading a recorded content into an editor, having a human watch the recording and mark regions for removal in the editor, and then exporting a new video with the unneeded segments removed. In essence, it is a

manual process requiring significant effort from a human editor to decide which portions of the recording are useful or not. An advanced method and system for automatically detecting and removing undesirable audio and video features in a media stream is described.

DETAILED DESCRIPTION

The systems and techniques described in this disclosure relate to a feature correction system that removes undesirable audio and video features from a media stream. The system can be implemented for use in an Internet, an intranet, or another client and server environment. The system can be implemented locally on a client device or implemented across a client device and server environment. The client device can be any electronic device such as a computer, laptop, mobile device, a tablet, a handheld electronic device, etc.

Fig. 1 illustrates an example method 100 for detecting and removing undesirable audio and video features in a media stream. The method 100 can be performed by the feature correction system.

The system determines 110 an undesirable audio feature in a media stream. The media stream can be any recorded content or a live streaming content. The media stream can be audio only or a video stream. The media stream can be either watched on a video streaming platform or on a local video watching platform on the client device. The undesirable features may include any type of audio regions that are not pleasant or not required for the listener of the media stream. For example, the system detects 112 a region that is relatively silent based on a threshold value. This is a simplest form of feature detection where the system applies the threshold value to a moving average of the waveform to detect regions that are relatively silent. Another example

can include, the system detecting 114 a region having a predetermined speech disfluency, vocalization, or utterance, e.g., umms, y'knows, ahhs, throat clearings, coughs, or repeated words. Simple feature detection for known filler utterances such as "um", "y'know","huh", "like", and "ah" may be predetermined in a straightforward manner. Complex feature detection for unfamiliar filler utterances such as variants in other languages, false starts where a speaker repeats a word, and repaired utterances where a speaker quickly corrects a previously spoken word, may rely on training artificial intelligence (AI) feature recognisers using neural networks and/or support vector machines.

Further, the system establishes 120 a start time and an end time for the undesirable audio feature. The system may store the start and end times on a local storage or on a cloud storage. The system then detects 130 corresponding video frames in the media stream that include the undesirable audio feature. The system finally performs 140 feature correction for the undesirable audio feature and/or corresponding video frames according to one or more predetermined techniques. The one or more predetermined techniques can be selected based on media stream. In an example, the system discards the undesirable audio feature or the system mutes 142 the undesirable audio feature. The system then analyses the changes in the video frames corresponding to the undesirable audio feature. The system checks 144 whether the changes in the video frames are above a threshold. The threshold can be determined by the system or can be manually entered by a human operator. The threshold value can be stored locally or on cloud storage.

If the changes in the video frame are above the threshold, the system compresses (146) the video. In another example, the system replaces the undesirable feature with a short silence

(mute), and compresses the corresponding video frame (timeline). In another implementation, if the changes in the video frame are above the threshold, the system changes the playback rate, i.e. fast-forward the video frame during undesirable audio features. If the changes in the video frame are below the threshold, the system does not compress the video frame, because for relatively non-static videos, the video stream does not appear to jump. Hence, the system aims to improve the experience of the listener by removing or shortening umms, ahhs, silences, repeated words, and other unnecessary disfluencies or delays in recorded content.

FIG. 2 is a block diagram of an exemplary environment that shows components of a system for implementing the techniques described in this disclosure. The environment includes client devices 210, servers 230, and network 240. Network 240 connects client devices 210 to servers 230. Client device 210 is an electronic device. Client device 210 may be capable of requesting and receiving data/communications over network 240. Example client devices 210 are personal computers (e.g., laptops), mobile communication devices, (e.g. smartphones, tablet computing devices), set-top boxes, game-consoles, embedded systems, and other devices 210' that can send and receive data/communications over network 240. Client device 210 may execute an application, such as a web browser 212 or 214 or a native application 216. Web applications 213 and 215 may be displayed via a web browser 212 or 214. Server 230 may be a web server capable of sending, receiving and storing web pages 232. Web page(s) 232 may be stored on or accessible via server 230. Web page(s) 232 may be associated with web application 213 or 215 and accessed using a web browser, e.g., 212. When accessed, webpage(s) 232 may be transmitted and displayed on a client device, e.g., 210 or 210'. Resources 218 and 218' are resources available to the client device 210 and/or applications thereon, or server(s) 230 and/or

web page(s) accessible therefrom, respectively. Resources 218' may be, for example, memory or storage resources; a text, image, video, audio, JavaScript, CSS, or other file or object; or other relevant resources. Network 240 may be any network or combination of networks that can carry data communication.

The subject matter described in this disclosure can be implemented in software and/or hardware (for example, computers, circuits, or processors). The subject matter can be implemented on a single device or across multiple devices (for example, a client device and a server device). Devices implementing the subject matter can be connected through a wired and/or wireless network. Such devices can receive inputs from a user (for example, from a mouse, keyboard, or touchscreen) and produce an output to a user (for example, through a display). Specific examples disclosed are provided for illustrative purposes and do not limit the scope of the disclosure.

DRAWINGS

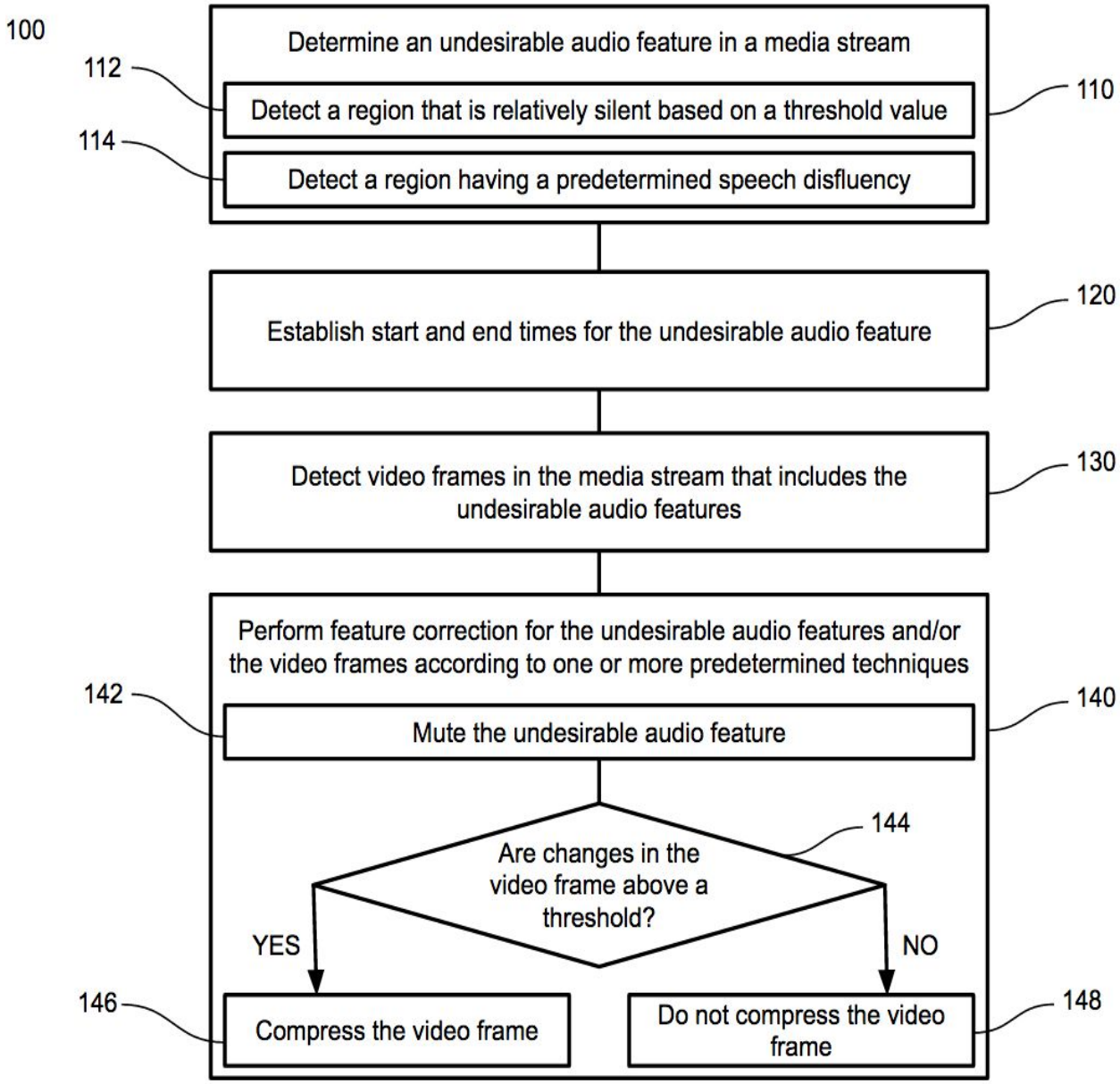

Fig. 1

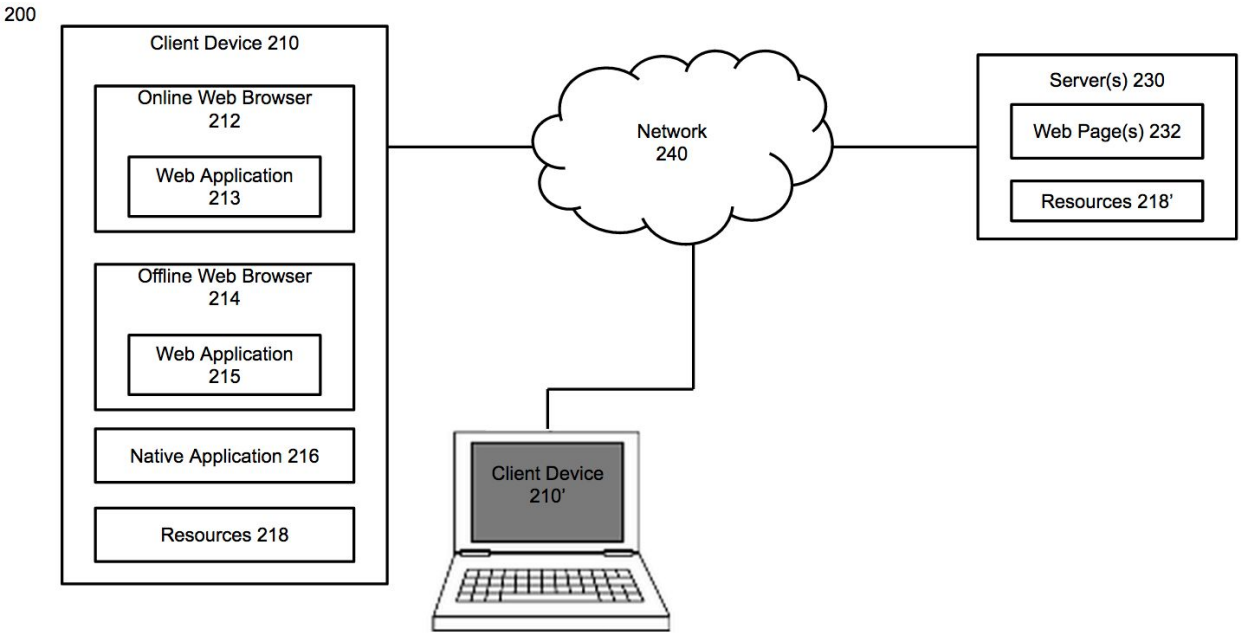
200
Client Device 210
Online Web Browser 212
Web Application 213
Offline Web Browser 214
Web Application 215
Native Application 216
Resources 218
Network 240
Server(s) 230
Web Page(s) 232
Resources 218'
Client Device 210'

Fig. 2