Technical Disclosure Commons

Defensive Publications Series

November 2019

Preprocessing Logic to Enhance Videotelephony Services During Suboptimal Network Conditions

Chinmay Dhodapkar

Preprocessing Logic to Enhance Videotelephony Services During Suboptimal Network Conditions

Abstract:

This publication describes processes and techniques, implemented on a computing device, directed at enhancing videotelephony services during suboptimal network conditions. In an aspect, a several-stepped process employs a machine-learned technique through which captured frames (*e.g.*, images captured by a videotelephony application) can be segmented and altered, such that an object of interest can be presented with a static background. The entirety of this process referred to herein as preprocessing logic, occurs prior to encoding.

Keywords:

Videotelephony services, video chat, video conferencing, internet protocol (IP) networks, internet protocol multimedia subsystem (IMS), communication services, poor network conditions, bitrate, encoder, bandwidth, image segmentation, static background

Background:

The Internet Protocol Multimedia Subsystem (IMS) is an architectural framework for delivering multimedia communication services (*e.g.*, voice services, videotelephony services) over Internet Protocol (IP) networks. Using IMS, live video, and audio streams can be transmitted and/or received by video-capable devices and systems, such as web cameras, mobile devices, videophones, videoconferencing systems, and telepresence systems. The process of transmitting video and audio streams involves the compression of a stream file by means of an encoder. The encoder generates a bitrate, which is directly proportional to the amount of motion in a captured scene. In other words, the more motion in a scene, the greater the data (bandwidth) that is required to transmit the stream. As a result, the quality of videotelephony services is sensitive to network conditions.

Under suboptimal network conditions (*e.g.*, when a network node or link is carrying more data than it can handle, resulting in queuing delay, packet loss, and/or the blocking of new connections), current IMS implementations follow one of two optimizations for handling network congestion situations during videotelephony calls. A first optimization is to downgrade the communication to "audio-only," removing the video stream.

A second optimization is to put a bandwidth cap on the bitrate of the encoder to decrease the bitrate of the stream. For example, the bitrate of the encoder could be decreased from 8,000 kilobits per second (kbps) to 400 kbps. Decreasing the bitrate of the stream ultimately results in lower resolution/quality for the whole frame (*e.g.*, the resolution could drop from 1920 x 1080 pixels to 640 x 320 pixels) and/or reduced frame rate (*e.g.*, the frame rate could drop from 60 frames per second (fps) to 30 fps), potentially causing the video stream to freeze on the display screen and/or audio-visual sync issues.

During poor network conditions, these optimizations rely on indiscriminately downgrading the captured frame to a lower resolution to account for the limited bandwidth. In many instances, however, the background of a scene, and not the object of interest (*e.g.*, a face), requires the highest bitrate allocation. Therefore, to enhance videotelephony services during suboptimal network conditions, a captured frame can be manipulated such that the object(s) of interest retains the greatest resolution, while the background in the captured frame can be replaced with a static image.

Description:

This publication describes processes and techniques, implemented on a computing device, directed at enhancing videotelephony services during suboptimal network conditions (*e.g.*, when a network node or link is carrying more data than it can handle, resulting in queuing delay, packet loss, and/or the blocking of new connections). In an aspect, a several-stepped process employs a machine-learned technique through which captured frames (*e.g.*, images captured by a videotelephony application) can be segmented and altered, such that the object of interest is presented with a static background. The entirety of this process, referred to herein as preprocessing logic, occurs prior to encoding.

An exemplary computing device, such as a smartphone, laptop, or tablet that can implement the preprocessing logic includes a display, sensors (*e.g.*, cameras, microphones, speakers), a processor, and a computer-readable medium (CRM). The CRM may include the operating system of the computing device and a machine-learned model (ML model). The ML model may be a standard neural-network-based model with corresponding layers required for processing input features like fixed-side vectors, text embeddings, or variable-length sequences. The ML model may be iteratively trained, off-device, to analyze frames and segment an object(s) of interest (*e.g.*, a face, a person) from the background. After sufficient training, the ML model can be deployed to the CRM. The CRM may also include a videotelephony application. The videotelephony application may utilize on-device sensors or peripheral devices to capture user input (*e.g.*, audio, video) when conducting a videotelephony call. Furthermore, the videotelephony application may contain an object detection algorithm that can detect an object of interest in a captured frame.

While conducting a videotelephony call during suboptimal network conditions, the preprocessing logic, in a first step, can detect an object of interest through the utilization of the native object detection algorithm in the videotelephony application. For example, a user (John) may be in a videotelephony call with his friend (Jane). Both John and Jane are situated in front of their respective computing devices, such that device sensors can capture images of their faces. Jane is conducting the call from her laptop at home. John, on the other hand, is conducting the call from his tablet computer at work. John's tablet computer experiences poor network conditions; thus, the preprocessing logic initiates the object detection algorithm to identify the object of interest, specifically John's face.

In a second step, the ML model can then analyze and segment the captured frames, such that the object of interest is segmented from the background. Continuing with the example, the ML model installed on John's device can segment John's face from the captured frames. In a third step, the background can be replaced with a static background (*e.g.*, wallpaper). For example, with John's face now segmented from the captured frame, the background can be replaced with a white background. After the preprocessing logic has segmented and altered the captured frames, the video data can be passed to the encoder for transmission. Concluding the example, Jane's laptop can receive the transmitted video stream and display John's face with a white background.

By implementing and executing the preprocessing logic in this manner, the object of interest can retain a high resolution even in suboptimal network conditions. Further to the above descriptions, a user may be provided with controls allowing the user to make an election as to both if and when they desire the captured frames to be segmented and altered. Furthermore, the user may be afforded the ability to select from a variety of static backgrounds.

In another example instance, a user (Mike) conducts a videotelephony call with another user (Dave). Both Dave and Mike are using smartphones to conduct the call. Additionally, both are situated in front of their respective smartphones, such that the device sensors can capture images of their faces. Initially, both Dave's and Mike's smartphones experience optimal network conditions. While walking down a sidewalk, however, Dave's device experiences sub-optimal network conditions. Figure 1 and Figure 2 illustrate Mike's smartphone displaying the generated video stream from Dave's smartphone.

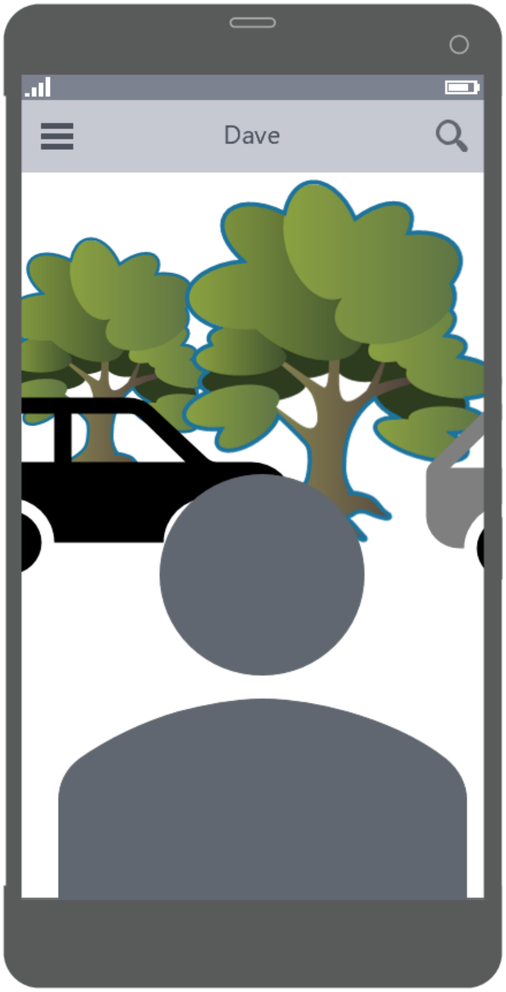

Figure 1

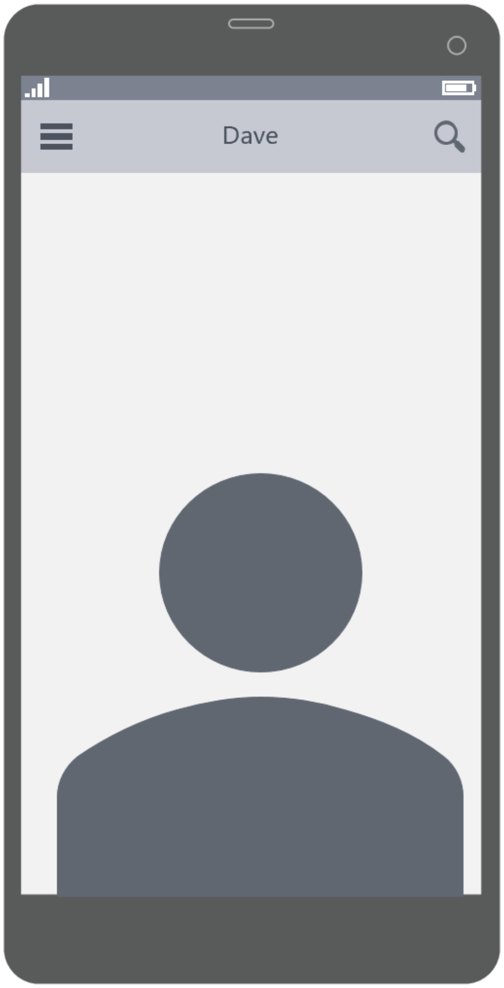

Figure 2

The video stream from Figure 1 contains the background because network conditions were initially optimal for Dave's device to encode all aspects of the captured frames. The video stream from Figure 2 illustrates a reproduced video stream that was generated at a later time under suboptimal network conditions.

As soon as Dave's smartphone experiences sub-optimal network conditions, preprocessing logic on Dave's device segmented and altered the captured frames. These captured frames had their backgrounds replaced with a static white wallpaper. As a result, the limited bitrate was allocated for encoding a high-resolution image of Dave.

In conclusion, the preprocessing logic implemented and executed, as described in this publication, may enhance videotelephony services when suboptimal network conditions have been detected.

References:

[1] Patent Publication: US20120082226A1. Systems and methods for error resilient scheme for low latency H.264 video coding. Priority Date: October 4, 2010.

[2] Patent Publication: US20070183662A1. Inter-mode region-of-interest video object segmentation. Priority Date: February 7, 2006.

[3] Patent Publication: US20120051631A1. System for background subtraction with 3D camera. Priority Date: July 23, 2010.