September 2021

AUTOMATIC DETECTION OF ON-SCREEN ADDRESSES IN VIDEO ENDPOINTS USED AS COMPUTER MONITORS

Rob Hanton

AUTOMATIC DETECTION OF ON-SCREEN ADDRESSES IN VIDEO ENDPOINTS USED AS COMPUTER MONITORS

AUTHORS:
Rob Hanton

ABSTRACT

When someone is making a video or audio call, manually typing in the address or the telephone number is both slow and error-prone. This is particularly true when the person is using a dedicated video endpoint with a contextual touchscreen keyboard. The situation is further exacerbated when video endpoints are employed as monitors in which the user interface (UI) of an endpoint when making calls frequently obscures or completely occludes the desktop screen. To address these challenges, techniques are presented herein that support, for situations in which a video endpoint is being used as a monitor, the video endpoint capturing text (when a call is placed) from the monitor feed to which it has access, automatically identifying any addresses or contact numbers, and providing such information as destinations that may be called either directly or once a user starts typing (e.g., through predictions). Aspects of the presented techniques may perform optical character recognition (OCR) during the analysis of text, may employ algorithms to calculate a confidence value for an identified address, and may employ one or methods to suggest identified addresses to a user.

DETAILED DESCRIPTION

When someone is making a video or audio call, manually entering the address or the telephone number is both slow and error-prone. This is particularly true when the person is using a dedicated video endpoint with a contextual touchscreen keyboard rather than a physical keyboard and where the address cannot be copied and then pasted. Such challenges are sufficiently problematic that online meeting platform vendors offer a wide range of methods to avoid the need for manual typing. Such methods include, for example, calendar integration, ‘one button to push’ (which, for convenience, may be referred to herein as ‘OBTP’) prompts, automatic callbacks when a meeting starts, the use of recent

call histories, and the automatic prediction of addresses based on previous calls and address book entries.

However, users cannot always leverage the methods that were described above. For example, a user's company may not have established calendar integration or OBTP, a user's company may employ endpoints from one vendor but the collaboration system of another, some users may schedule meetings in such a way that addresses are not automatically captured, or an individual may simply have a pattern of work where users message each other with the addresses to a call. Additionally, some users may predominantly dial addresses manually. Even in enterprises in which the work flow normally allows for better alternatives to address such issues, on occasion, users may still need to manually dial addresses.

The challenges that were described above are exacerbated when video endpoints are employed as monitors. Video endpoints often have high-quality screens and so are frequently used as monitors to save both cost and desk space. However, under such circumstances not only does the user need to enter an address by hand on a touchscreen keyboard, but the user interface (UI) of the endpoint when making calls frequently obscures or completely occludes the desktop screen. Such an arrangement may lead to users not being able to see the address that they are copying or needing to rearrange window locations to make such information visible.

To address the different challenges that were described above, techniques are presented herein that support, for situations in which a video endpoint is being used as a monitor, the endpoint capturing text (when a call is placed) from the monitor feed to which it has access, automatically identifying any addresses or contact numbers, and providing such identified information as destinations that may be called either directly or once a user starts typing (e.g., through predictions).

Many users of hardware endpoints often choose to employ their endpoint as a monitor when it is not in a call. As such, when a user is seeking to make a call and they do not have access to helpful tools such as calendar integration, the last action that they take prior to using the on-screen touch panel to manually dial the call is highly likely to be opening on their screen a message, meeting invitation, or email containing the address that they will be calling along with other associated information such as, if relevant, a meeting

password and a personal identification number (PIN). Since in a high percentage of cases such a screen will be available to the endpoint, this can be used to automatically intuit the user's intention and mitigate the need for them to manually type the address.

As noted previously, a first element of the techniques presented herein encompasses an endpoint identifying on-screen text. Aspects of this element will be discussed in the following narrative.

In circumstances where a dedicated video endpoint system is being used as a monitor, the endpoint may perform optical character recognition (OCR) on the video feed that is being displayed to identify text. Such an OCR process may be run only at times where the user interacts with the endpoint controls in some implementation-dependent way (such as, for example, when a user brings up the controls, when the user explicitly presses a 'call' button, etc.) or it could be run periodically when the endpoint is in use and not in a call (e.g., every five seconds). An implementation may elect to limit such activity to when the endpoint is not in another call.

The OCR detection may be run locally on the endpoint or snapshots (or even the video feed itself) may be sent to a remote system, such as the cloud, where OCR and analysis may be performed. Since the endpoint will generally be performing the OCR and analysis when the endpoint is not in another call, resources (from, for example, a central processing unit (CPU) or a graphics processing unit (GPU)) that are normally used for decoding and displaying remotely-received video will generally be idle and thus would be available for such analysis without requiring increases in the endpoint's capabilities or bill of materials (BoM).

For privacy reasons, an implementation may provide an option for users and/or administrators to disable any analysis of local cabled video, or limit such an analysis to specific times (e.g., when a 'call' button is pressed, when a specific 'get address from my screen' option is selected, etc.). This may be of particular value if the device is sending images or text to a separate server (e.g., in the cloud or elsewhere) for analysis, but even if the processing is taking place locally some users may prefer if the device did not process their local video in any way or unless explicitly invoked.

As noted previously, a second element of the techniques presented herein supports analyzing text and a display screen for addresses and other information. Algorithms, along

with various implementation options, for this element will be discussed in the following narrative.

The simplest algorithm for analyzing text and a display screen encompasses identifying portions of the text that match one or more sets of supported address formats (such as, for example, email style, telephone numbers, etc.). However, an algorithm may also incorporate textual or non-textual information to estimate a confidence value that the address is actually something to which the participant has a high probability of wanting to place a call.

The factors that might increase or decrease the confidence value for an associated body of text might include:

- The presence of words such as 'call' that suggest that an artifact is an address for calling;
- The presence of words such as 'email' that suggest that an artifact is not an address for calling;
- The presence of specific wording that is automatically generated by tools (such as calendar integration facilities) for meeting invitations for certain video conferencing services;
- The presence of a domain name (e.g., abcxyz.com) that is associated with video or audio conferencing; and/or
- An assessment that the OCR process itself has correctly identified all of the letters.

The algorithm may be rules-based or it may be derived from machine learning that is trained on many snippets of text containing an address that is designated as being intended as callable or not.

While it may require more complexity and processing power, an implementation may also include other information from a video stream in its estimation of a confidence value. Such other information might include, for example, whether the text containing the address is in a top-most, focused pane or whether previous frames suggest that the text was typed by the user themselves.

An implementation may also incorporate other information that is not gathered from the video stream for calculating a confidence value, such as whether an address has

ever been called in the past or whether it shares a domain with an address that has been called.

Along with addresses an implementation may also look to identify passwords and PINs that are associated with placing a call, particularly if the algorithm recognizes specific automatically-generated wording that is used in calendar integrations.

As noted previously, a third element of the techniques presented herein supports utilizing captured addresses. For example, an implementation may involve one or more methods through which it can automatically suggest identified addresses to a user. In one instance, each method may have a confidence threshold that is associated with it such that only suggestions that meet or exceed the confidence threshold may be suggested. Generally, higher confidence thresholds may be set for methods involving more intrusive thresholds.

Methods in which a suggestion may be presented to a user may include, for example:

- For cases where the OCR process is running periodically, superimposing an OBTP suggestion on the screen.
- Displaying explicit options when the user touches a 'call' button.
- Displaying a 'suggested' category of entries alongside recent calls.
- Providing autocomplete suggestions once a user starts to manually enter an address that matches the initial part of one of the suggestions.

An implementation may support a number of the methods, applying one method or another depending upon which confidence thresholds are met. For example, an implementation might capture the text that is depicted in Figure 1, below.

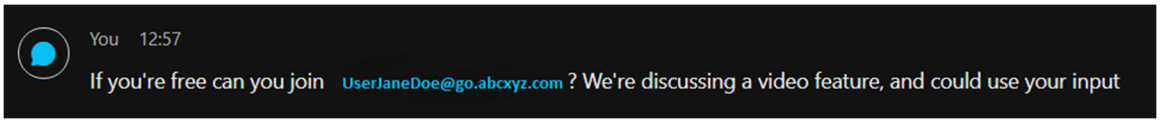

Figure 1: Illustrative High Confidence Example

Referring to Figure 1, above, based on the presence of associated words such as 'join' and the recognition of the 'abcxyz.com' domain, an implementation may calculate a high confidence value of 0.91 (out of a maximum value of 1.0) and choose to overlay an explicit 'call UserJaneDoe@go.abcxyz.com' button on the screen for the user to select.

Alternatively, an implementation might capture the text that is depicted in Figure 2, below, and identify a potential address. However, without any associated verbiage and with an unrecognized domain name it might calculate a much lower confidence value of 0.28.

Figure 2: Illustrative Low Confidence Example

Referring to Figure 2, above, the calculated confidence value might be too low to cause it to overlay an explicit 'call user2@m.altranet.com' button on the screen. However, if the user were to select a 'call' button the implementation might include 'user2@m.altranet.com' in the list of recent and suggested addresses. Alternatively, an implementation might include 'user2@m.altranet.com' as an autocomplete suggestion should the user enter 'u' as the first letter.

In addition to the features and capabilities that were described and illustrated in the narrative that was presented above, aspects of the techniques presented herein support a number of alternatives. As just one example, during the analysis of text and a display screen for the identification of addresses and other information (as was described above) in addition to the presence of particular words (such as, for example, 'call,' 'join,' etc.) other attributes such as formatting, previous usage, etc. may also be incorporated.

In summary, techniques are presented herein that support, in situations where a video endpoint is being used as a monitor, the video endpoint capturing text (when a call is placed) from the monitor feed to which it has access, automatically identifying any addresses or contact numbers, and providing such information as destinations that may be called either directly or once a user starts typing (e.g., through predictions). Aspects of the presented techniques may perform OCR during the analysis of text, may employ algorithms to calculate a confidence value for an identified address, and may employ one or methods to suggest identified addresses to a user.