January 2023

Video Preview Image Selection Based on Prior User Interactions

D Shin

Video Preview Image Selection Based on Prior User Interactions

ABSTRACT

Many websites and applications that enable users to view videos include a feature that lets a user watching a video hover on the timestamp bar to see preview images of the content at or near that timestamp. However, selection of preview images that is unaware of the video content can cause the thumbnails to be non-representative and unappealing to the user. This disclosure describes the use of crowdsourced information on video pause actions performed by various users to select preview images for different parts of a video. An engagement prior model is built on the crowdsourced information. A temporal histogram of the video based on prior user pause actions enables intelligent preview image selection, e.g., using a gradient ascent algorithm. The thumbnails thus selected can provide a more satisfactory preview experience for the user. The temporal heatmap can be updated and preview image selection repeated as more viewers view the video and perform pause actions.

KEYWORDS

- Video thumbnail
- Preview image
- Video pause
- Pause event
- Temporal histogram
- Gradient ascent
- User engagement model
- Engagement metrics

BACKGROUND

Many websites and applications that enable users to view videos include a feature that lets a user watching a video hover on the timestamp bar to see preview images of the content at or near that timestamp. This helps the user to conveniently scroll through the video without stopping content streaming.

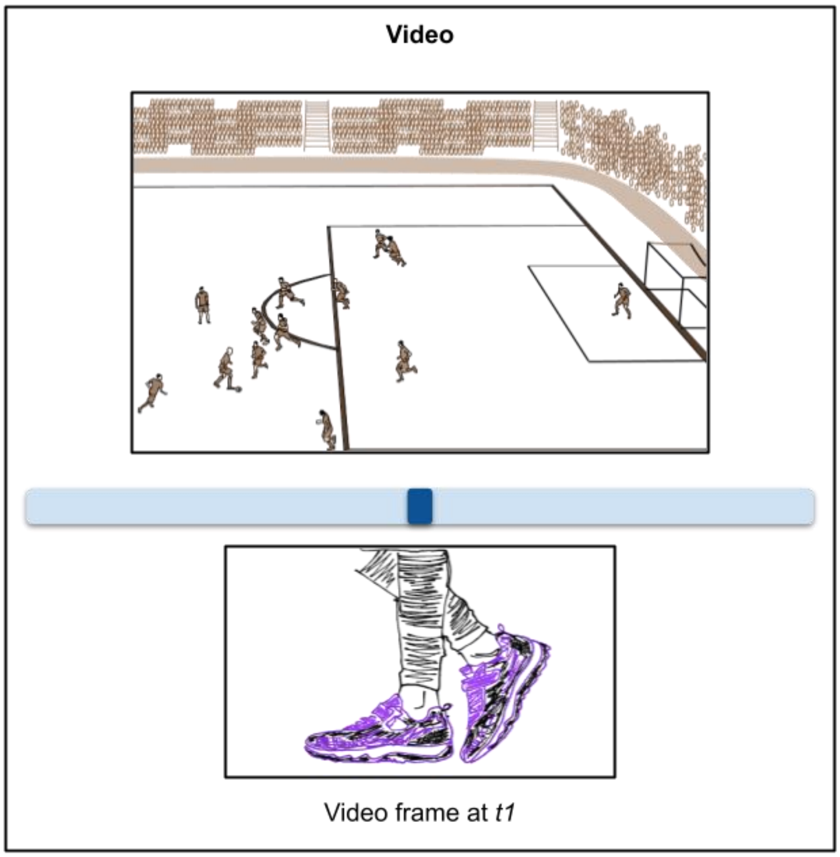

Fig. 1: Naive selection of preview image

A naive way of choosing the preview images is to perform uniform preview image pooling across the video. In this method, an image snapshot of the video is obtained at regular intervals, e.g., every two seconds across the entire video. For any timestamp that the user hovers on, the nearest image snapshot is shown as the preview image. Fig. 1 illustrates a video of a football match, with a video frame at time t1 being selected as a preview image. As can be seen in fig. 1, the naive selection causes an uninteresting image (a closeup of a player's legs) to be selected as the preview. The preview images selected based on periodicity or other techniques that are unaware of content saliency at different timestamps can be unappealing to users since these are essentially random images selected without taking into account user interests, rather than being based on prior user engagement. This can be a dissatisfactory experience for users trying to locate the content they would like to see near a particular time location of the video.

DESCRIPTION

This disclosure describes the use of crowdsourced information on video pause actions performed by users to select preview images for different parts of a video. An engagement prior model is built on the crowdsourced information. The model enables intelligent preview image selection.

For a given video, pause actions performed on the video by various users are determined, with appropriate permissions from the viewing users. Only the video timestamps of the pause events are gathered at runtime; no user-specific context information is obtained. The pause timestamps are used to generate a temporal histogram. The temporal histogram has a multimodal structure given that a single video can have many interesting moments throughout at which various viewing users may pause the video.

To obtain preview images, an initial set of timestamps is identified. For example, for a video of length M seconds, N-second uniform image pooling is performed to identify M / N uniform timestamps for corresponding preview images. This conventional preview image generation step is used to initialize preview image selection.

The preview images are updated using a fine-tuning technique. For each preview timestamp, a gradient ascent algorithm is applied on the temporal histogram obtained from the crowdsourced pause events. The gradient ascent algorithm searches for “local modes” (pause timestamps that were the most common among prior viewers). After each gradient ascent from a preview timestamp, the timestamp for the preview image is shifted closer to local mode, based on the step size chosen. The gradient ascent is repeated until a convergence criterion is met. Optionally, the gradient ascent design can incorporate some stochasticity to ensure that it does not get stuck in a secondary local peak.

The total number of gradient ascent operations performed is $k\_1 + k\_2 + \ldots k\_(M / N)$, where k_i denotes the total number of ascent steps taken per preview image. Also, the operations over k_i are parallelizable, such that if the number of processing clusters matches the number of timestamps (M / N), the time complexity is $min(k\_1, k\_2 \ldots, k\_(M / N))$.

After the optimization, the resulting timestamps are used to fetch corresponding images from the original video. These are used as preview images seen by the end user in the scrolling user interface.

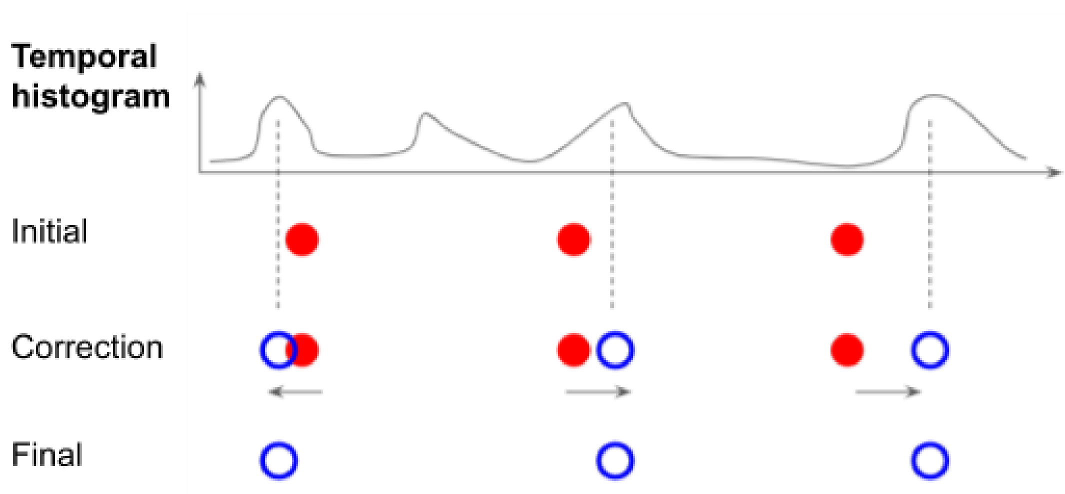

Fig. 2: Naive selection of preview image

Fig. 2 illustrates an example of preview timestamp optimization, per techniques described herein. The temporal histogram feature can have many valleys that may not align with the initial preview timestamps (uniformly separated) since the valley locations depend on where the interesting “pause-able” content is within the video.

The correction via gradient ascent is performed for each timestamp. An important constraint is that a correction should not shift the timestamp beyond a pre-defined upper bound of T seconds. This ensures that the user is not shown a preview that is time-distorted in a way that does not accurately represent the content in the actual video at that time.

Optionally, backward corrections that select a preview image from an earlier timestamp than the regular interval can be penalized more than forward corrections that select an image that occurs after the regular interval. This is because when the user selects a timestamp using the preview, the video play. With forward corrections, the user is likely to find content that generated the preview within a short interval, e.g., in the next second or so. However, if a prior timestamp is selected for preview, the user may not see the preview image in the video that plays.

As seen in Fig. 2, initial (regularly spaced) timestamps are adjusted using gradient ascent. The final optimized timestamps are shown in the last row as minor shifts from the original timestamps. A key property of the preview images generated using the final timestamps is that these preview images are more likely representative of high-engagement content near the user-selected timestamp, which helps the user to make a more targeted search.

The temporal histogram is a dynamic feature that is periodically updated as more users view the video and perform pause actions. The histogram topology changes over time. To update preview image selection, the gradient ascent technique can be selectively re-run by comparing the L2-distance between the normalized histograms of a first time (when a correction was performed) and an updated histogram. An L2-distance that exceeds a predefined threshold indicates that how viewers define interesting image content in the video has changed and can trigger a re-run of the correction scheme.

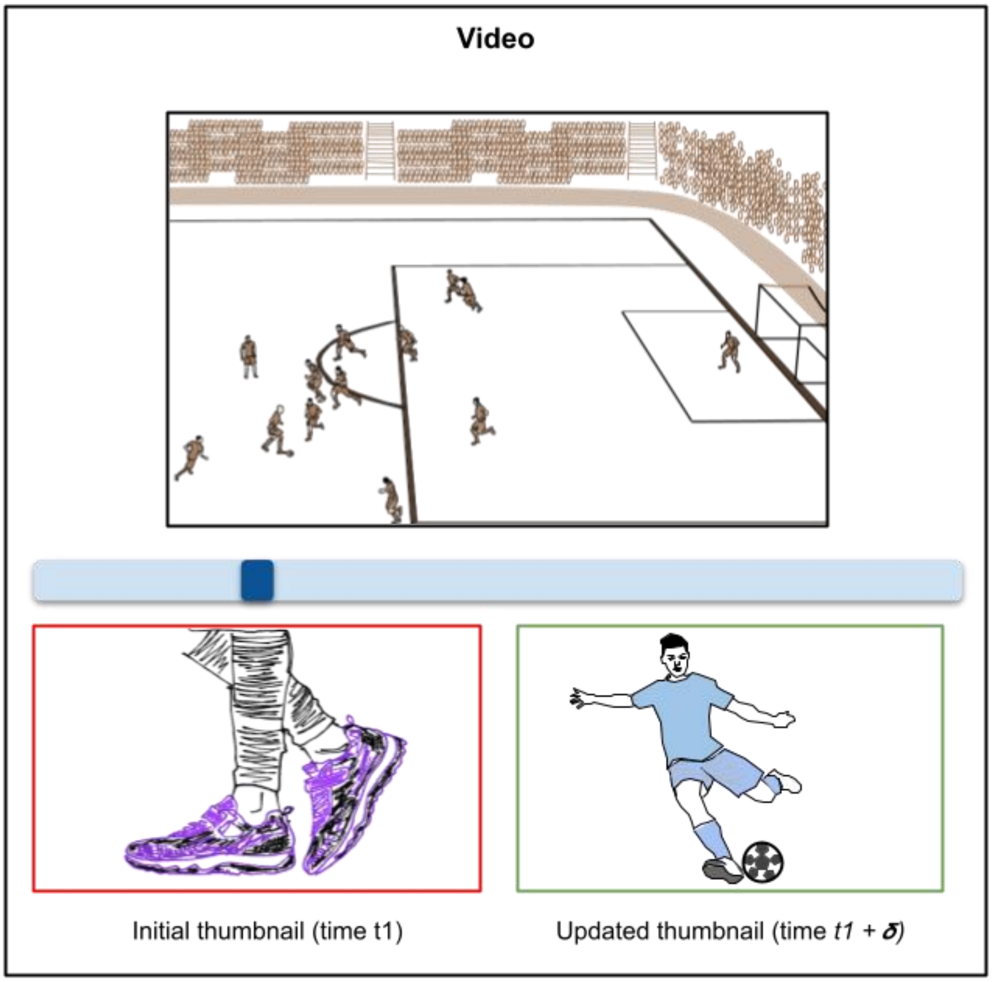

Fig. 3: Initial and corrected preview images

Fig. 3 illustrates the video of Fig. 1 with a corrected thumbnail. The initial thumbnail (left-side, shown in red border) is unsatisfactory as it depicts uninteresting action. With the techniques described herein, when the user scrolls to time t1, a preview image from time t1 + $\boldsymbol{\delta}$ is shown (right-side, shown in green border). The thumbnail image depicts the action of a player kicking a ball (which is more interesting than a closeup of a player’s legs) and is selected based on prior user pause actions, as described above.

Further to the descriptions above, a user may be provided with controls allowing the user to make an election as to both if and when systems, programs or features described herein may enable collection of user information (e.g., information about a user’s interaction with a video including pause actions. social network, social actions or activities, profession, a user’s preferences, or a user’s current location), and if the user is sent content or communications from a server. In addition, certain data may be treated in one or more ways before it is stored or used, so that personally identifiable information is removed. For example, a user’s identity may be treated so that no personally identifiable information can be determined for the user, or a user’s geographic location may be generalized where location information is obtained (such as to a city, ZIP code, or state level), so that a particular location of a user cannot be determined. Thus, the user may have control over what information is collected about the user, how that information is used, and what information is provided to the user.

CONCLUSION

This disclosure describes the use of crowdsourced information on video pause actions performed by various users to select preview images for different parts of a video. An engagement prior model is built on the crowdsourced information. A temporal histogram of the video based on prior user pause actions enables intelligent preview image selection, e.g., using a gradient ascent algorithm. The thumbnails thus selected can provide a more satisfactory preview experience for the user. The temporal heatmap can be updated and preview image selection repeated as more viewers view the video and perform pause actions.