April 12, 2019

AN IMPROVED METHOD FOR INTELLIGENT SCREEN LAYOUT

Robert Barton

Jerome Henry

Maik Seewald

Mojan Mobasser

AN IMPROVED METHOD FOR INTELLIGENT SCREEN LAYOUT

AUTHORS:
Robert Barton
Jerome Henry
Maik Seewald
Mojan Mobasser

ABSTRACT

Techniques are described for a mechanism to intelligently change layouts of a video conference or other collaborative meeting based on a variety of possible inputs (e.g., age of the shared screen, user preference, use of white boards, audience gaze, word matching, etc.). The mechanism resizes the shared screen for a collaborative meeting using an algorithm that estimates the current attention to the screen versus meeting participant intercommunications. The goal of the mechanism is to right-size the shared content so that it does not interfere with the live video conversation and to optimize the overall experience for meeting participants.

DETAILED DESCRIPTION

Most modern collaboration meetings involve multiple participants connected through video as well as a screen sharing component. It is common to mix the layout of remote video feeds with the screen that is being shared. One expectation is that most participants prefer to view the shared screen rather than view the other participants. However, this assumption has certain drawbacks. Among them, the screen sharing portion of the conference tends to dominate the viewing space making it difficult to see the other remote participants. Some implementations have a function in which remote participant windows can be increased (and the shared screen portion decreased), but this function is manual and its functionalities are often limited. It is a regular occurrence that the conference will digress from the subject presented on the slide, yet the inactive slide continues to dominate the viewing space, reducing the overall impact of the collaborative experience.

When a collaborative meeting is being conducted, an objective is to maximize the immersive nature of the meeting. This objective is often missed because the shared content dominates the screen and the video interactions are minimized. In larger conferences, the video portion of the participants is almost entirely obfuscated by the collaborative meeting shared content. Manual adjustment is difficult because the shared slide does need to be the main focus at some times. In most meetings, there are times when the conversation between participants should be dominant and the shared content should be minimized. However, dynamic readjustment of the layout is not possible under these conditions.

For example, if a slide has remained on the screen for a long time, but the conversation has drifted away from the slide, it means the slide has become stale and is no longer active to the current conversation. At this point, it would be better to reduce the shared slide portion and increase the video participants sections, thus improving the experience for all. Today, it is common that someone will have to "stop sharing" in order to have the video participants maximized.

This proposal presents a novel (and improved) mechanism to right-size the viewing space for both the video participants as well as the shared screen. In particular, the mechanism provided by this proposal outlines five separate methods that can be implemented to optimize the layout of the screen in which the method invoked will depend on the situation and the shared context. It should be noted that the method invoked on each endpoint (e.g., for each participant in a call) can be different, meaning that each endpoint in a call may have a different layout scheme, depending on how various conditions are met, as discussed in further detail below.

Method 1: Reducing the shared screen size using a half-life aging timer mechanism

For a first method, when content is shared, each slide that is presented is given an initial freshness value by the collaboration system (e.g., an initial freshness value of 100 indicates a new slide) and a timer starts for the slide. The freshness of each slide is a function of time windows called the slide half-life.

By way of analogy, compare the concept of radioactive-decay in which highly radioactive substances decay according to a known half-life, which is different for each substance. Within the context of collaborative meeting slides, when a new presenter begins to show a slide, the slide is assigned a default half-life time and an initial freshness value, which is some predetermined amount of time that is default for all new slides and presenters.

Until the timer reaches the first half-life marker, the slide remains at the same default size. However, after the first half-life expires, the shared content window is reduced in size by 50% (or any other determined amount), indicating that its freshness has decreased by a proportional amount. After another half-life expires, the window is again reduced in size by a further 50% (or any determined amount). The freshness value would be 25% of the original (e.g., it would have a freshness value of 25). This process can be repeated until a slide is reduced to a predetermined minimum size. Figure 1, below illustrates example details associated with a standard reduction mode that can be used for dynamically resizing a shared screen.

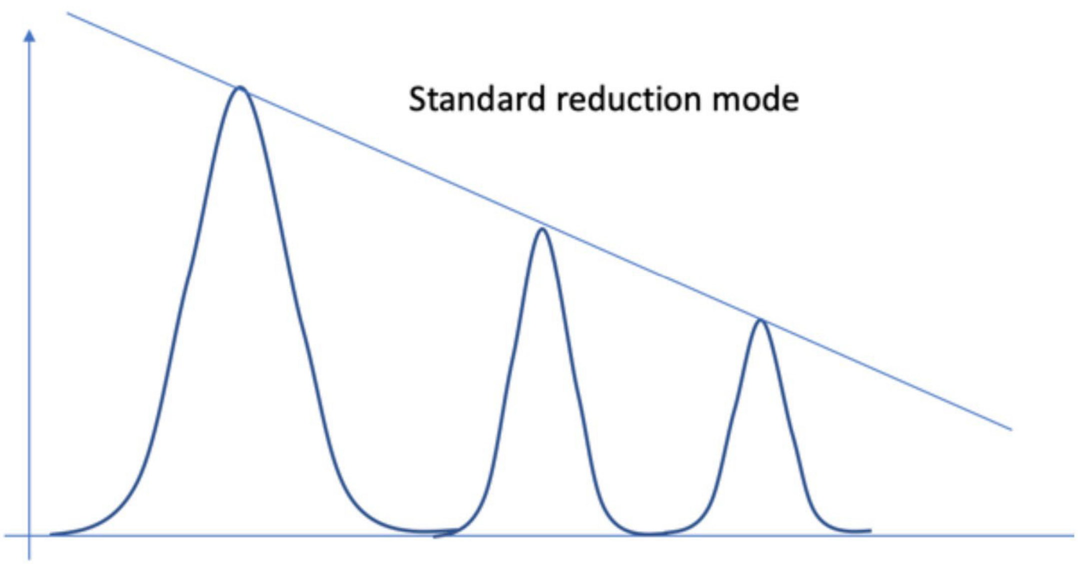

Figure 1

For the first method, as the size of the shared content is reduced, the size of the video participants is increased. Increasing the size of the video portion follows the same logic. Video endpoints that have been active (and not on mute) will be given a higher freshness value and will be given a proportionally larger area of the screen. Thus, this process of assigning a half-life to screen layouts and using it to dynamically resize a screen layout provides a novel method for resizing the size of shared content as well as video participants.

Method 2: Automatic resizing based on learned user preferences

For a second method, the initial method, as discussed above, is employed and a 'user preferences' setting records the interaction of a local user and the collaborative meeting screen.

In one example, each time the slide dimension is reduced, a small option can appear with the option "stop reducing now." One possible technique to implement this is to have the user reach out and touch the shared frame of the screen to stop it from reducing in size any further. For such a technique, a "double-tap" could start the process once again.

As automated reduction continues, when the user manually resizes the slide section (to smaller or larger), the system records a tuple consisting of: (slide automatic size, manually re-dimensioned size, screen specs [dimensions/resolution]). In one example, an option to save the settings appears such as, for example, "save dimensions preferences."

In another example, the user action is recorded to influence the later Gaussian curve. In an example implementation, the Gaussian curve is instead a skew normal distribution with shape 0 (e.g., mean/location 0 and scale 1), as illustrated in Figure 2. It should be noted that a skew normal distribution resembles the Gaussian function but also determines the location of the mean (while Gaussian assumes position 0). This parameter is used to document the point in time (as shown on the x-axis in Figure 2) where the use freezes the size. The skew normal also includes a scale (omega, the height of the curve, which is the size of the slide section in this case) and a shape (which allows the skew, and that is used as shown below to change the rate at which the slide section continues to reduce).

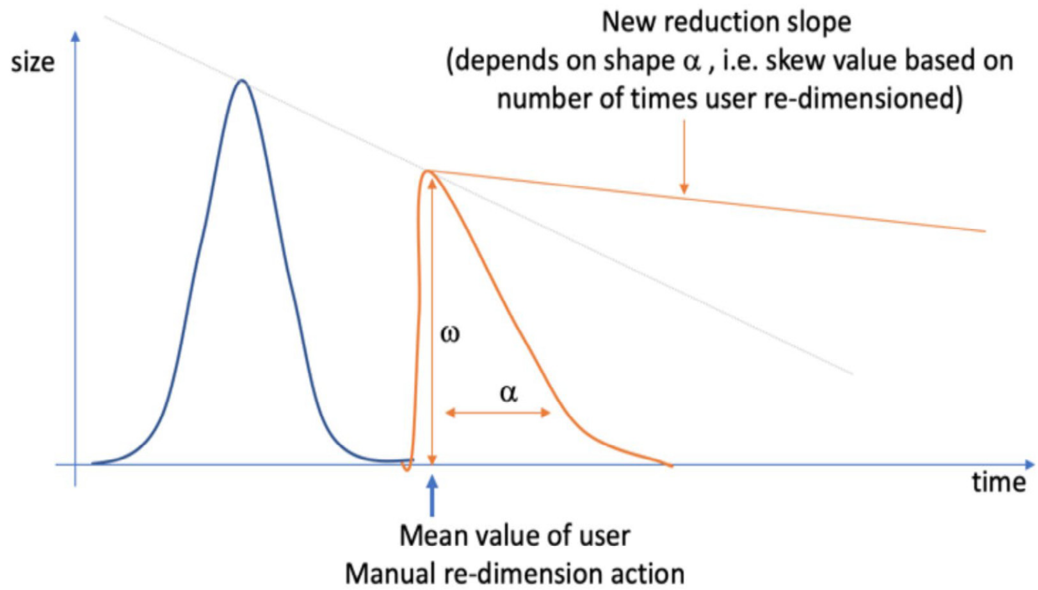

Figure 2

Each time the user interrupts the slide dimension change, the shape alpha is increased. Each additional time increases the scale (omega). The location of the mean (alpha) is moved to the mean of the user manual resizing value. The result is that the slide tends to move faster toward the user preferred size, and to stay longer at the user preferred size, before slowly continuing to be reduced.

Method 3: Observing usage of white boards as an indication of slide relevancy

For a third method, the system can observe usage of collaborative tools, such as white boards, as an indication of whether attention to a slide is no longer relevant. For this method, as movements on the shared screen are recorded, detection of a moving mouse over slide items (or pen actions) causes the Bayesian curve to invert (thus increasing the slide size at high rate). Similarly, but with an inverse logic, detection of movements (mouse, pen) on an area other than the slide area cause that area to increase its size rapidly. The increased slope can either be linear (e.g., max size within 0.5 seconds) or use the above skewed normal curve logic, where alpha (the speed of increase/decrease) is directly related by the number of actions detected within a target interval (e.g., each tens of a second).

In one example, the system will detect if other collaboration tools are being used, such as a collaborative white board. This is a case in which a member of the audience gets up and begins to white-board something, thus drawing attention away from the shared content frame. In this example, the shared frame is resized in the same manner as above.

Method 4: Tracking words or expressions as an indication that a slide is still fresh

For a fourth method, the telepresence system's (the local unit/collaboration system) Natural Language Processing (NLP) system is used to track words or expressions as an indication that a slide is still fresh. In this mode, an Optical Character Recognition (OCR) service detects and records words on the slide (such as "as you can see on this slide"). As the main speaker's speech is recorded, a voice analysis module compares the words that are said to the list of words recorded for the slide.

Various existing techniques can be used to avoid false positives. For example, "is" may be on a slide and said often, but such a word is structural construct of the speech and not necessarily relevant to the slide content. Thus, such grammatical articulation words or expressions can be ignored. Similarly (but with an opposite logic), bags of words may store 'visual attraction catcher' expressions, such as: "as you can see," "in the upper right part," etc.

Several implementations are envisioned. In one of example, identification of individual words causes the slide size to stop reducing. In another example, detection of expressions (e.g., consecutive words) causes the slide to re-increase in size. In yet another example, when voice keywords are used indicate slide freshness, the half-life timer is extended.

Method 5: Gaze detection using Computer Vision to determine audience gaze

For a fifth method, gaze detection using Computer Vision is used to determine if the audience is looking at shared content or if they are attempting to squint to see other people on the video conference. Various existing techniques that allow for the detection of gaze direction may be utilized in this mode.

For example, a counter can record eye movements and the time spent over each area of the screen over an interval. An example of time spent over different areas of a screen are illustrated below in Figure 3.

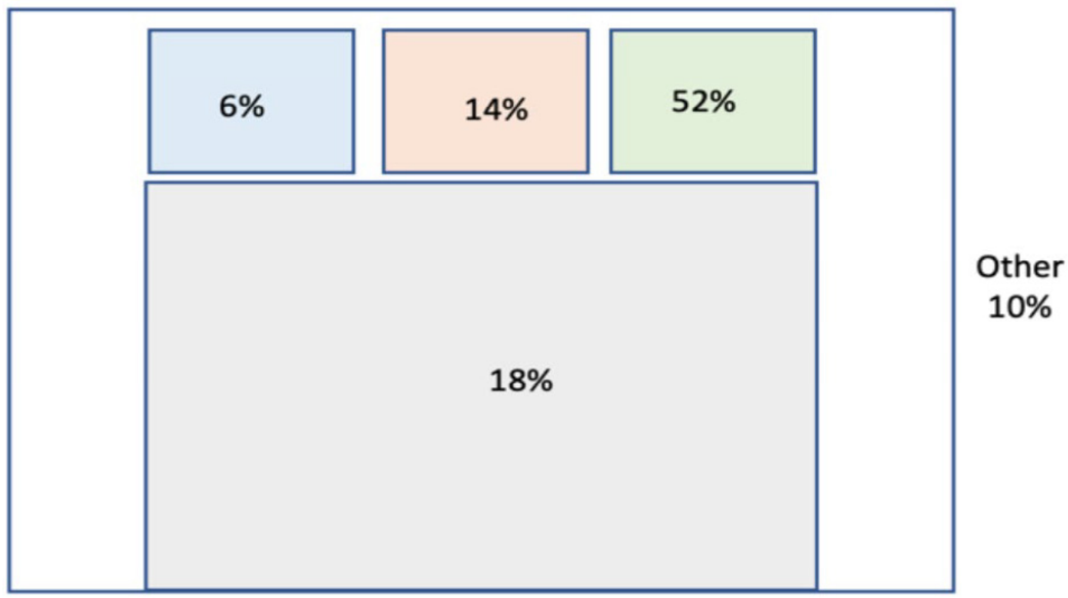

Figure 3

As the time spent over the slide decreases (compared to a previous interval), the system dynamically reduces the size of the slide area (as per the above techniques) and progressively increases the size of the elements that are watched more often. Figure 4, shown below, illustrates an example in which slide areas may be resized utilizing this method.

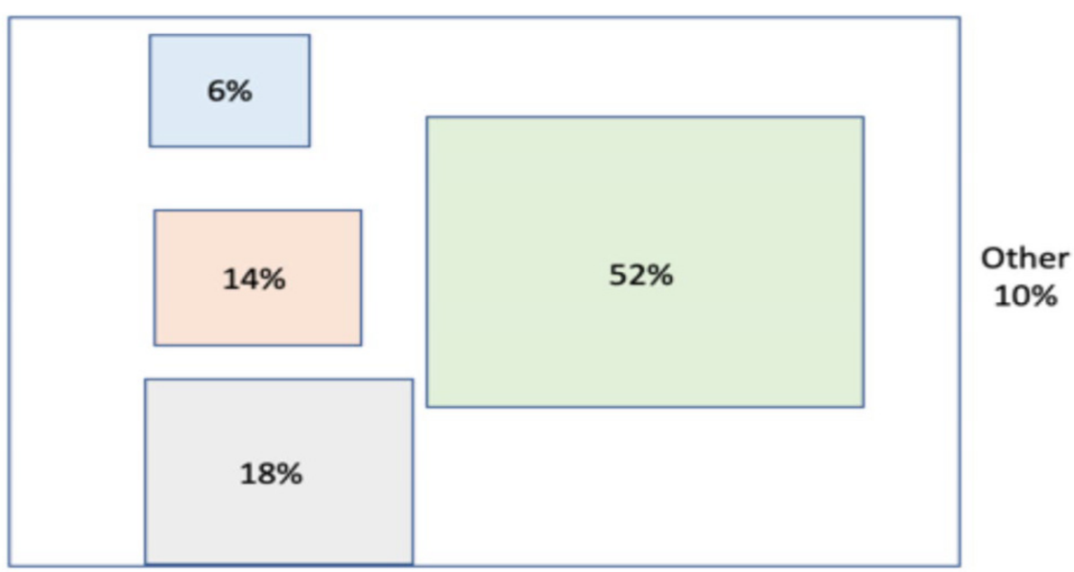

Figure 4

Special bias can be given to specific areas. For example, slides that are not looked at may reduce slower than pictures of participants that do not receive looks. Additionally, a sliding mechanism can also push the windows of lesser interaction to the side of the screen. For example, if a user primarily watches the center and upper right of a screen, this position will receive the active windows while elements that may be on the right, but not watched, will slowly slide to the lower left pat of the screen. One advantage of this mode is that participants for which the video is off (dark screen) will slowly be reduced and moved to a position where they do not occupy a significant part of the screen real-estate.

Overriding the Automated Layout Mechanism using Reinforcement Learning

There are times when an automatic size reduction may be incorrectly applied to shared content. If the size of the shared content is incorrectly reduced too soon, a collaborative meeting assistant can be employed to reset the layout to the original size.

For example, a user may provide an input such as, "Okay tool, please increase the size of the shared content." Additionally, content may need to be fixed in-place. This can be performed either with a soft-key on the collaborative meeting system or with specific hand gestures. For example, using a method of sign language an indication can be provided that a particular frame should be fixed at the correct size. Ultimately, these actions become an input to a reinforcement learning model.

If the system incorrectly resizes the frame too early requiring an intervention, the previous layout is restored and the half-life of the slide is immediately reset and the freshness is doubled from the past value. This now becomes an application of reinforcement learning. While the system has predicted the optimal size of the frame and got it wrong, the user input to the slide becomes part of the mechanism to train future predicted values. In this manner, the reinforcement learning engine records the premature shrinkage of the frame and then incrementally increases the default half-life for all slides in the presentation.

Consider the following example for implementing the core technique of a half-life based screen layout mechanism using Bayesian Inference. While a default half-life value is expected to work in many cases, the mechanism is greatly improved by dynamically learning and adjusting the half-life to each presentation and presenter. It is assumed that the time spent on slides will follow the normal distribution (Gaussian distribution model). Based on this model, an initial default half-life is chosen at the mean value of the Gaussian curve. The next step for this proposal is to learn and optimize a better suited half-life value for each presentation.

To implement this technique, a model using Machine Learning (ML) may be used to predict the most ideal half-life, and then it is adjusted (by skewing the Gaussian curve) as more slides are shown and the dwell time of each slide is measured. An application of Bayesian Statistical Inference is used for this example. The Bayesian model used for this example will combine the prior probability of a slide's half-life and with new learning as slides are changes to generate a posterior probability using the well-known Bayes Theorem, shown below as:

$$P(h \mid d) = [P(d \mid h)\ P(h)] / P(d)$$

For this example: $P(h \mid d)$ is the posterior distribution (that is, the half-life prediction model after application of Bayes model); $P(h)$ is the prior knowledge (the Gaussian distribution used as a starting point for all presenters and presentations); $P(d \mid h)$ is the observed behavior; and $P(d)$ is a constant for the example. Example details associated with measured dwell time on slides is shown below in Figure 5.

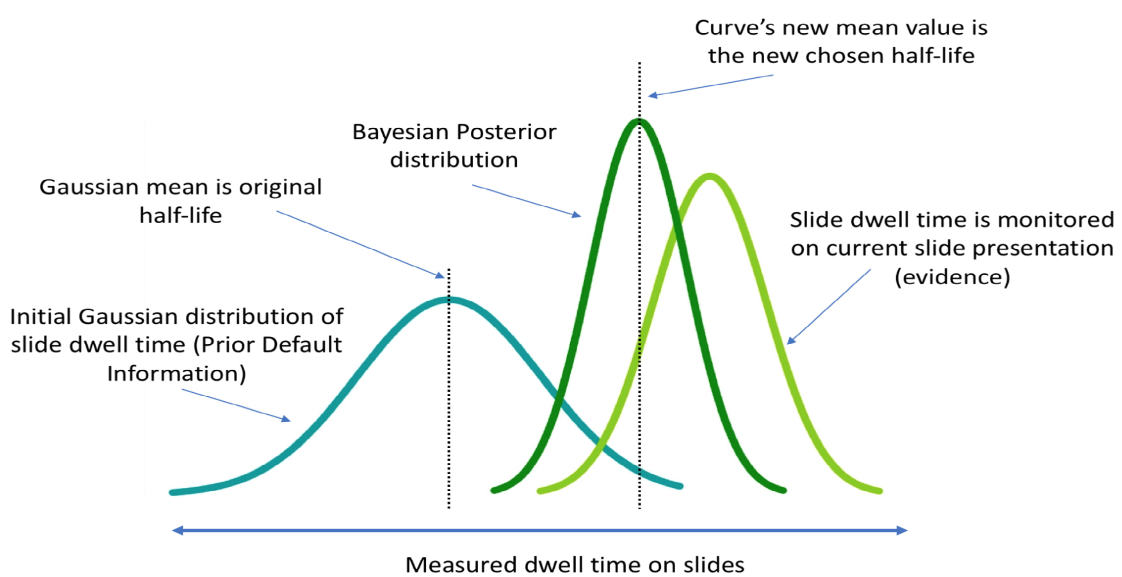

Figure 5

Thus, as the slides are changed in the presentation, the rate of change is monitored and captured in real-time. The Bayesian model is applied and a new half-life is learned (the mean value of the new Bayesian posterior probability). For example, if the observed rate of change is fast, the half-life will be reduced. If the rate of change is slow, the half-life is increased. Thus, the model adapts to each presentation's slide change cadence.

In some instances, slides may be incrementally revised. In one example implementation, any change in a slide frame could be considered as refreshing the half-life timer. Thus, if there is an animation involved in a slide, such animation would reset the half-life timer. This is why the ML techniques provided by this proposal become so powerful. Such techniques do not care if there are new slides versus animations involved; the learning mechanism can easily adapt to both scenarios. In yet another example implementation, a damping mechanism could be injected into the Reinforcement Learning model as a feature to extend the half-life in order to address any potential sawtoothing of displayed slide size that may be encountered during system operation.

In summary, techniques are described for a mechanism to intelligently change layouts of a video conference or other collaborative meeting based on a variety of possible inputs (e.g., age of the shared screen, user preference, use of white boards, audience gaze, word matching, etc.). The mechanism resizes the shared screen for a collaborative meeting using an algorithm that estimates the current attention to the screen versus meeting participant intercommunications. The goal of the mechanism is to right-size the shared content so that it does not interfere with the live video conversation and to optimize the overall experience for meeting participants.