RoIRTC: TOWARD REGION-OF-INTEREST REINFORCED REAL-TIME VIDEO COMMUNICATION

Shuoqian Wang
SUNY Binghamton
swang130@binghamton.edu

Mengbai Xiao
Shandong University
xiaomb@sdu.edu.cn

Yao Liu
Rutgers University
yao.liu@rutgers.edu

ABSTRACT

In this paper, we propose a region-of-interest (RoI) reinforced real-time communication system, RoIRTC, for improving the quality of videos delivered in real-time communication. RoIRTC uses a novel RoI magnification transformation for spatially adapting the camera-captured video frame. To automatically detect the RoI, it intelligently leverages a deep-learning-based saliency prediction model without affecting the video collector's processing throughput or the encoder's efficiency. Evaluation results based on actual remote learning videos show that RoIRTC that performs RoI magnification can improve the median PSNR by 2.6 dB compared to the naive WebRTC implementation. Compared to an approach that mimics the "background blur" scheme used in many real-time communication systems, RoIRTC can also improve the median PSNR by 4.2 dB.

1. INTRODUCTION

Since the start of the COVID-19 pandemic, people have increasingly been using real-time communication for their daily activities. At work space, face-to-face meetings were turned into online meetings; In schools and universities, in-person instructions were replaced with remote instruction sessions. A popular framework for real-time communication today is WebRTC [1], used by video conferencing applications such as Google Meet. Despite the increased demand of these video conferencing applications, many households have limited access to the Internet, or can only access the Internet at low and unreliable network bandwidth. The bandwidth issue is not unique for real-time video communication. In traditional on-demand video streaming applications, existing works have proposed to use solutions including tiling [2, 3] and cropping [4, 5]. However, these approaches cannot be used in the real-time scenario. For tiling, each tile needs to be encoded independently, while under the real-time scenario, computing the target bitrate and encoding parameters for each tile and encoding is a time consuming job. For cropping, important information may be missed. While it is possible to "rewind" in the on-demand video streaming setting, it is not feasible for real-time communication. Some existing video codecs support APIs for encoding different regions of the frame at varying quality, e.g., the "Emphasis MAP" supported by the Nvidia NVENCODE API [6]. However, they may result in rate control violations and are not appropriate for real-time communication under limited available network bandwidth.

Today, popular video conferencing applications such as Zoom and Google Meet have incorporated features such as background blurring and virtual background that can help reducing the bandwidth needed for video transmission. However, similar to cropping, these approaches can blur out or cover up otherwise important visual content. In addition, they can be distracting to users [7].

To improve the quality of real-time video communication when only limited network bandwidth is available between the sender and the receiver of the video stream, in this paper, we present our design of a region-of-interest reinforced real-time communication system, RoIRTC. RoIRTC is designed based on characteristics of the human visual system: only a small region on the retina called fovea has the highest density of photoreceptors, resulting in highest visual acuity; while the peripheral vision has decreased visual acuity [8]. This allows us to optimize real-time video communication by first identifying important areas of the video frame and then spatially adapt the frame to devote more pixels to these areas. For example, consider a synchronous online learning scenario where the teacher is showing flashcards to the students, we expect the students to focus their attention to the flashcards and the teacher's actions, while details of the rest of the classroom may not be as important. We refer to the important areas where we expect the viewers to focus their attention on as the "Region-of-Interest" (RoI) area. Unlike existing works that pre-encode the RoI areas as tiles in different resolutions, RoIRTC performs real-time spatial adaptation to the captured frame so that more pixels in the frame are devoted to the RoI. This means more bits are used for the RoI area in the encoded frame, allowing the RoI to be transmitted in higher quality than the rest areas in the frame. Compared to using RoI-encoding that may be supported by video codecs, the spatial adaptation approach used by RoIRTC also has the benefit of smooth quality transitions across the frame. In our design, RoIRTC also flexibly allows the RoI to be determined either manually by feedback from the video receiver or automatically via a deep-learning-based saliency detection model. Overall, this paper makes the following contributions:

- We propose a novel RoI-magnification transformation approach that can spatially adapt the captured frame and devote more pixels in the frame for encoding the RoI region,

allowing the RoI to be delivered in higher quality.

- We design an automatic ROI selection approach via deep-learning-based saliency detection and make sure that the automatic RoI selection does not negatively impact the video encoder's operation.
- We implement RoI magnification within the WebRTC framework (hence the name RoIRTC) and integrate it with a deep-learning-based saliency model for RoI selection.
- Evaluation results show that RoIRTC can significantly outperform other baseline approaches in the visual quality metrics including PSNR, SSIM, and VMAF.

2. BACKGROUND AND RELATED WORK

2.1. Real-Time Video Communication and WebRTC

Real-time video communication has very stringent latency requirements: the end-to-end latency of the video stream must be low to allow interactive video communications (e.g., less than 150 ms one-way for verbal communications [9]). In addition, the processing (e.g., encoding, frame transformation, etc.) throughput of the video stream must match the frame rate of the video stream, e.g., 30 frames-per-second (FPS). The low-latency requirement makes real-time video communication much more challenging compared to traditional on-demand video streaming. WebRTC is the de facto standard open-source framework to support real-time communication. For rate control under limited network bandwidth, WebRTC estimates the available network bandwidth and uses it as the target bitrate of the video encoder. The encoder returns the encoded frame as well as the corresponding quantization parameter (`qp`) used for encoding. The `qp` value is monitored by an adaptation controller at WebRTC, who decides how and if it is necessary to adapt the frame resolution and frame rate. For example, if the target bitrate cannot be achieved even after setting `qp` to its maximum value, WebRTC will reduce the encoded frame resolution. Rate control in WebRTC, however, adapts the encoding of video frames as a whole without considering the varying importance of different areas in the frame. As a result, under limited network bandwidth, the quality of the visual content may be severely reduced. In comparison, our RoIRTC framework can improve the visual quality by considering the saliency of different areas of the frame.

2.2. Region-of-Interest (RoI)

Given that only a small area of the video frame may be of interest to the viewer and that human eyes only have a small fovea region with the highest visual acuity, existing works have proposed region-of-interest streaming systems. For example, Ryoo et al. [10] proposed a foveated video streaming service. It uses a tile-based approach, pre-encoding 16x9, a total of 144 "grid cells" in six base resolutions. During streaming, cells of different resolution levels are selected and downloaded based on the real-time gaze data collected about the viewer. Pang et al. [11] proposes an interactive region-of-interest video streaming system for devices with small display sizes. This system allows the viewers to select arbitrary RoIs that will be transmitted in high resolution. Similar ideas of RoI streaming are also used in the scenario of 360-degree video streaming, where a viewer can only observe part of the omnidirectional content, e.g., [12, 13]. On the other hand, these systems focus on the on-demand streaming scenario, where the video tiles are pre-encoded and can be requested during playback.

3. DESIGN OF RoIRTC

RoIRTC improves the visual quality of real-time video communication by using a spatial-adaptive approach. Figure 1 shows the overall workflow of RoIRTC. Here, we use an image of a checker board placed on a table to represent a frame to be transmitted. Content in the dashed line box with white background represents the workflow at the video receiver, while content in the solid line boxes with gray background represents the workflow at the video collector. RoIRTC has two main components: (1) automatic/manual region-of-interest (RoI) selection for identifying the RoI area of the video frame; and (2) frame transformation for improving the visual quality of the RoI by devoting an increased number of pixels on the encoded frame for representing the RoI area. Note here that data about areas outside of the RoI, e.g., the classroom background, is still transmitted to the receiver's side, but in lower quality than the RoI area. This is different from naive approaches that only transmit the RoI area or blur the areas outside of the RoI.

3.1. Frame Transformation in RoIRTC

For ease of explanation, we represent a frame as a rectangle on a 2D coordinate system xy. The center of the frame is located at the origin of the coordinate system $(0, 0)$, and the rectangle is bounded by $x = \pm 1$ and $y = \pm 1$. We represent the transformation function as $T(x, y) = (f(x), f(y))$, where, (x, y) is the 2D coordinates of a pixel on the "transformed frame", and $f(x)$ and $f(y)$ represent the coordinates of the pixel in the original frame that is being transformed to coordinate (x, y). Given that the rectangle is bounded by $x = \pm 1$ and $y = \pm 1$, we have $|x| \leq 1, |y| \leq 1$. When no frame transformation is applied, we have $T(x, y) = (f(x), f(y)) = (x, y)$. We also must ensure that after magnification of the RoI, all content is still contained within the frame and no content is popped out, i.e., $|f(\pm 1)| = 1$. Given a transformation function, we need to verify how much the content is magnified. To do so, we can represent the magnification along one axis, e.g., the x-axis, as magnification $= \frac{\partial}{\partial x} f^{-1}$.

3.1.1. Transformation Function

We select the following transformation function that results in a smooth transition of magnification amount across the frame:

$$T(x, y) = (f(x), f(y)) = \Big(\tan(\frac{\pi}{4} \cdot x), \tan(\frac{\pi}{4} \cdot y)\Big) \quad (1)$$

We refer to this transformation as "**RoI Magnification Transformation**". To calculate the magnification amount

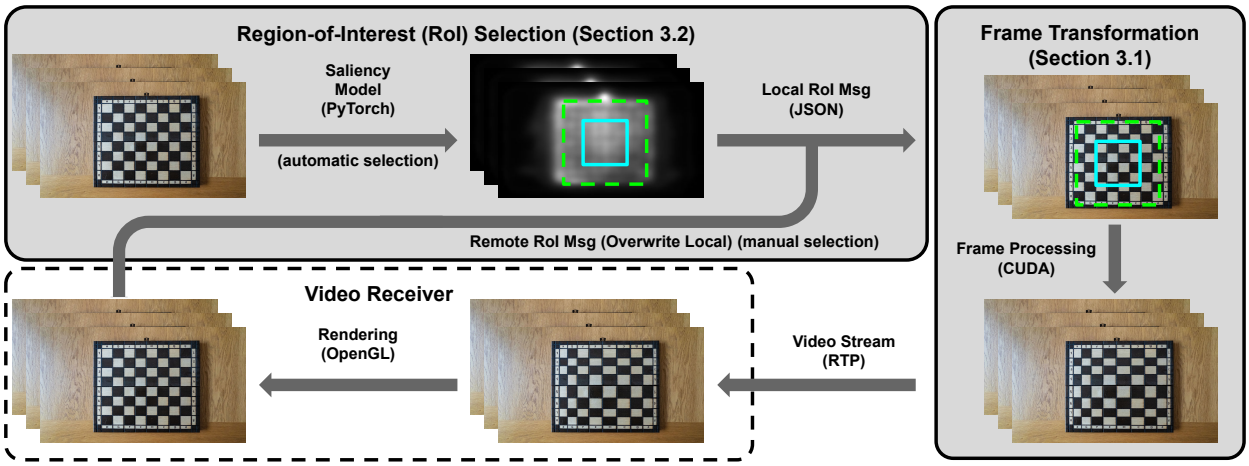


Fig. 1. This figure shows the overall workflow of RoIRTC. The dashed line box represents components at the video receiver; while the solid line boxes (with gray background) represent components at the video collector.

across the frame, we first obtain the inverse of the transformation function as follows:

$$\begin{bmatrix} f_x^{-1} \\ f_y^{-1} \end{bmatrix} = \begin{bmatrix} \frac{4}{\pi} \cdot \arctan x \\ \frac{4}{\pi} \cdot \arctan y \end{bmatrix} \tag{2}$$

The inverse of the transformation function can be used to calculate the coordinates of the transformed pixels given its coordinates in the original frame. We can then use the Jacobian matrix to calculate the amount of magnification:

$$\text{magnification} = \begin{bmatrix} \frac{\partial f_x^{-1}}{\partial x} & 0 \\ 0 & \frac{\partial f_y^{-1}}{\partial y} \end{bmatrix} = \frac{16}{\pi^2 \cdot (1+x^2) \cdot (1+y^2)} \tag{3}$$

With this magnification function, the center of the frame is at $(0, 0)$ and is magnified the most. Not all of the area is magnified. Only areas within $x \in (-0.523, 0.523)$ and $y \in (-0.523, 0.523)$ are magnified, i.e., magnification > 1.0. The remaining areas near the border of the frame will appear to be reduced. We refer to these areas that are magnified as "**Region of Magnification**".

3.1.2. Transformation Calculation

The analysis above is based on the assumption that the RoI is at the center of the frame. In practice, this may not always be the case. To account for more general scenarios, we propose a "sub-area-only" approach. This approach selects a sub-area within the frame that contains the RoI plus a margin area outside of the RoI. This sub-area needs to be big enough so that pixels within the RoI are indeed magnified after the transformation, while pixels in the margin area may appear to be reduced. We know from Equation (3) that for the magnification to be greater than 1, the width and height of the sub-area needs to be $\frac{1}{0.523} \approx 1.912$ times the width and height of the RoI.

Suppose we have a RoI at time t, we can construct a sub-area of the frame such that RoI is located in the center of the sub-area. We then represent this sub-area as x_t, y_t, w_t, h_t, where (x_t, y_t) is the upper left corner of the area, and w_t, h_t are the width and height of the area, respectively. The center of RoI, (which is also center of the sub-area), can thus be represented as:

$$\begin{bmatrix} o_x \\ o_y \end{bmatrix} = \begin{bmatrix} x_t + \frac{w_t}{2} \\ y_t + \frac{h_t}{2} \end{bmatrix} \tag{4}$$

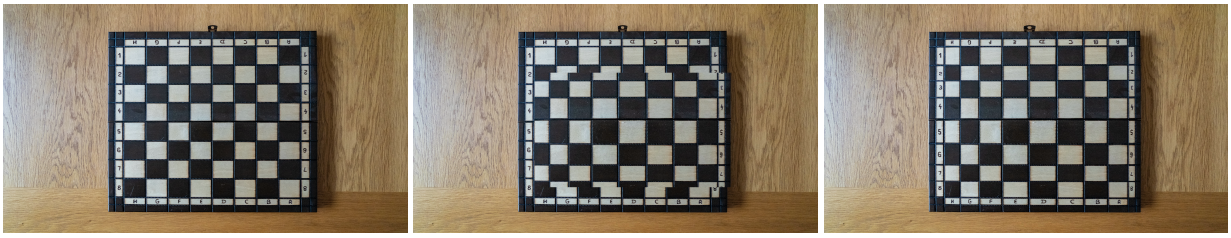
Given a point (x, y) within the sub-area, we can perform the re-centering and change the origin of the frame from $(0, 0)$ to (o_x, o_y) as below:

$$\begin{bmatrix} x_{ot} \\ y_{ot} \end{bmatrix} = \begin{bmatrix} \frac{2}{w_t} \cdot (x - o_x) \\ \frac{2}{h_t} \cdot (y - o_y) \end{bmatrix} \tag{5}$$

Conceptually, this can be interpreted as re-centering the sub-area to $x \in [-1, 1]$ and $y \in [-1, 1]$. For pixels whose $x_{ot} \in [-1, 1]$ and $y_{ot} \in [-1, 1]$, that is, pixels within the sub-area, we can then apply the RoI magnification transformation as follows:

$$\begin{bmatrix} x'_{ot} \\ y'_{ot} \end{bmatrix} = \begin{bmatrix} f_x \\ f_y \end{bmatrix} = \begin{bmatrix} \tan(\frac{\pi}{4} \cdot x_{ot}) \\ \tan(\frac{\pi}{4} \cdot y_{ot}) \end{bmatrix} \tag{6}$$

During our implementation, we uncovered a complication associated with the sub-area-only approach: if transformation is only performed within the sub-area, there appears to be clear "edges" (i.e., discontinuities) at the sub-area boundaries. An example is shown in Figure 2(b). Such "edges" may negatively affect the performance of the video encoder. To address this problem, we expand the ranges where transformation is performed along the x and y axes separately. That is, we perform transformation f_x for all $x_{ot} \in [-1, 1]$ (regardless of y_{ot} values) and transformation f_y for all $y_{ot} \in [-1, 1]$ (regardless of x_{ot} values). As a result, the transformed frame data still appears to be smooth, as is shown in Figure 2(c).



(a) The original, unprocessed frame (b) "Sub-area-only" (c) Our final proposed approach

Fig. 2. If we only perform the transformation to a selected sub-area that centers the RoI (Figure (b)), then clear "edges" (discontinuities) appear at the boundaries of the sub-area. Our final proposed approach (Figure (c)) results in smooth frame without "edges" while magnifying the RoI area by devoting more pixels on the transformed frame to it.

3.2. Automatic RoI Selection in RoIRTC

RoIRTC can support both manual and automatic approaches for determining the region of magnification. **Manual RoI selection** is straightforward as in Figure 1: the video receiver transmits the RoI via messages to the video collector, and these messages have a higher priority compared to automatically-selected RoI. For **automatic RoI selection**, RoIRTC leverages deep-learning based saliency detection models to analyze the captured frame and identify one RoI area in each frame for magnification.

Simply using the RoI center identified by the deep-learning-based saliency model for frame transformation, however, can result in frequent, short-range RoI changes. When combined with RoI magnification, these can make it more difficult for the encoder to perform motion estimation across frames to exploit inter-frame compression. As a result, the encoding performance may suffer. Next, we describe our automatic RoI selection approach that addresses these challenges.

The RoI changes can be the result of high-frequency, small movements of the objects, e.g., because people's body or objects held in people's hand cannot stay steady. In this case, simply following the RoI changes can cause the video encoder to suffer as it becomes more difficult to find matching areas across frames to exploit inter-frame compression. To this end, we use a low-pass filter to filter the high-frequency vibration of the RoI. On the other hand, RoI changes caused by stable, long-range movement of objects in the video, should be followed. In this work, we treat movement whose range is within the latest sub-area bounding box as short-range "high-frequency" noise, while long-range movements are followed immediately. Assuming that based on saliency model output, the RoI center's coordinate is $(\hat{x}, \hat{y})$, and that the latest RoI center's coordinate used by the frame transformation module is (x, y), we can calculate the new coordinates of the RoI center x', y' for use by frame transformation as:

$$x' = w_x \cdot \hat{x} + (1 - w_x) \cdot x \tag{7}$$

$$y' = w_y \cdot \hat{y} + (1 - w_y) \cdot y \tag{8}$$

where w_x and w_y are calculated weights. Taking the horizontal direction (i.e., x) for example, if the normalized distance between the predicted RoI and last RoI is smaller than 1.0 (recall that only pixels with $x \in [-1, 1]$ are transformed in the horizontal direction), we treat these movements as noises and select the weight: $w_x = (\frac{\pi}{4} \cdot (\hat{x} - x))^2$. If the movement range is greater than 1.0, we need to follow as soon as possible. Thus, we set the weight as: $w_x = 1.0$.

Besides spatial changes, we also need to account for variations in saliency prediction time. Suppose the average processing time for RoI prediction is $\hat{T}$, and the actual model processing time is T, we calculate w_x as:

$$w_x = \min\Big(\frac{T}{\hat{T}} \cdot (\frac{\pi}{4} \cdot (\hat{x} - x))^2, 1\Big) \tag{9}$$

4. IMPLEMENTATION

We implement RoIRTC as a fork of the open-source WebRTC framework[1]. Figure 1 shows the overall workflow of RoIRTC. As video frames are captured at the video collector's side, a saliency model processes the captured frames and returns information about the predicted RoI. Here, the RoI is illustrated using the cyan box in solid lines. The RoI is surrounded by the margin area as illustrated using the green box in dashed lines. Given the RoI, RoIRTC performs spatial adaptation of the frame via CUDA for fast frame transformation. The processed frame will then be sent to the receiver via the real-time protocol (RTP) [14]. At the video receiver side, received frames need to invert the transformation to display the correct video content to the user. To do so, we implement OpenGL shaders to render the frame from the RoI-magnified frame.

At the video collector's side, our RoIRTC implementation includes three modules: our modified WebRTC running as the application process, a server running a deep learning model for saliency prediction, and a signaling server for facilitating the communication between WebRTC and the saliency model. For saliency prediction, we used UNISAL [15]. We tested the model on a desktop computer with Intel i7-7700K GPU and NVIDIA GeForce GTX 1080 GPU. When input frames are in the resolution of 1280×960, UNISAL can only achieve approximately 9 frames-per-second (FPS). When the input resolution is reduced to 960×640, 640×480, and

[1]RoIRTC source code is available at `https://github.com/bingsyslab/RoIRTC`.

480×320, the frame rate is increased to 17, 29, and 33, respectively. This indicates that we can substantially reduce the per-frame saliency prediction time by reducing the input frame size. Based on this profiling results, we chose to use 640×480 as input to the saliency prediction model.

5. EVALUATION

In this work, we focus our evaluation of RoIRTC on the synchronous online education scenario. We selected and downloaded 9 remote learning videos from YouTube. These videos represent various remote learning scenarios where teachers used whiteboards, books, hand writing, and other teaching instruments for teaching. We decode all videos into the `.y4m` format, which is a uncompressed format, to be used as the virtual webcam stream as captured video frames.

During the experiments, we record frames of the input stream and use them as the ground truth. For visual quality comparison, we also record every playable frame at video receiver side and match them with their corresponding original frames recorded at the video collector's side. Recording all frames in its uncompressed YUV420 format is I/O intensive. Instead, we selectively record only the original frames at the video collector whose timestamps assigned by Chromium are even. For frames delivered via RoIRTC, we restore the frame using OpenGL shaders before visual quality comparison. We focus our comparison on the RoI area as this is the area where we expect the users to focus their attention on. For visual quality comparison, we use three metrics: peak signal-to-noise ratio (PSNR), structural similarity (SSIM), and video multi-method assessment fusion (VMAF) [16].

We use the `naive WebRTC` implementation as our baseline. For RoIRTC, we evaluate three configurations: RoIRTC with RoI magnification (`R-mag`); RoIRTC with blur (`blur-only`); and RoIRTC with hybrid blur and RoI magnification (`hybrid`). For the RoIRTC with `blur-only` configuration, we identify RoI of the frame as with other RoIRTC configurations. However, no transformation is performed for the RoI. Instead, we blur the areas outside of the margin area in the frame. This is similar to the "background blur" approaches used in existing video conferencing platforms such as Zoom and Google Meet. The `hybrid` configuration performs RoI magnification within the margin area and blurs the area outside of the margin area. We note that as areas in the frame are being blurred, much of the visual information is lost. If the user focuses her attention on the blurred area, the visual quality is extremely low. This can severely affect the user's quality of experience. For fair comparison, we compare `R-mag` with `naive WebRTC`, and compare `hybrid` with `blur-only`. Due to space limit, we only report our evaluation results under 2.5 Mbps available network bandwidth.

5.1. Visual Quality Results

Figure 3 reports the distribution of these visual quality results from different comparative approaches. We focus on the visual quality of the RoI in this figure. Results show that for RoIRTC with `R-mag`, its median PSNR is 2.61 dB higher than the `naive WebRTC` implementation, its median SSIM is 0.05 higher, and the median VMAF is 10.29 higher. We know that WebRTC framework chooses a target bitrate for use by the encoder using statistics collected about the network condition. When content outside of the margin area is blurred, much visual details are removed from the frame. Under the same bitrate budget, content within the margin area can now be encoded in higher quality, e.g., more high frequency content can be retained in lossy encoding. Therefore, it is not surprising that for both `WebRTC` with `blur-only` and RoIRTC with `hybrid`, the visual quality of frames are higher than both `naive WebRTC` and RoIRTC with `R-mag`. Comparing `WebRTC` with `blur-only` against RoIRTC with `hybrid` R-mag+blur, we find that the median PSNR, SSIM, and VMAF of the `hybrid` approach is 4.2 dB, 0.025, and 12.08 higher than `blur-only`, respectively. This indicates that our RoI magnification approach can effectively improve the visual quality of the RoI in all testing scenarios.

In addition to the RoI quality comparison, Figure 4 compares the visual quality of the sub-areas (i.e., RoI plus the margin area) used in transformation. Note that since our comparative approaches include blurred areas, we choose not to compare the quality of the full frame. In our RoI magnification approach, the margin area includes areas that are reduced. Even so, the overall visual quality of RoIRTC with `R-mag` is still better than `naive WebRTC` in all but one scenarios and metrics. In addition, the median visual quality results of RoIRTC with `hybrid` are better than `blur-only` in all scenarios. That is, even with the margin area where the visual content may be reduced, using RoI magnification transformation still generates better visual quality.

6. CONCLUSION

To improve the visual quality of real-time communication under limited available network bandwidth, in this paper, we proposed RoIRTC. RoIRTC uses a novel region-of-interest (RoI) magnification transformation for spatially adapting the captured video frame and devoting more bits in the encoded frame to the RoI area. It further uses a deep-learning-based saliency model for automatically detecting the RoI. We implemented RoIRTC based on the WebRTC framework and compared it against both the naive WebRTC framework and an approach that only blurs the non-RoI content. Results show that our proposed RoIRTC can deliver the video frame in higher visual quality compared to these baseline approaches.

Acknowledgments. We appreciate constructive comments from anonymous referees. This work is partially supported by NSF under grants CNS-2200042 and CNS-2200048.

7. REFERENCES

[1] "WebRTC," https://webrtc.org/.

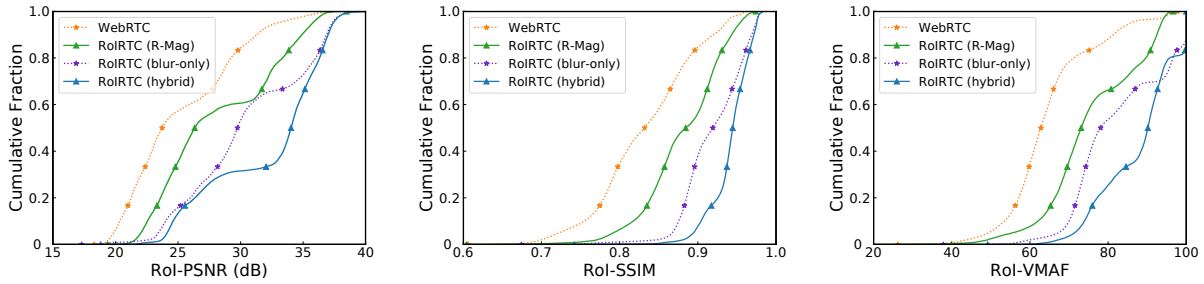


Fig. 3. RoI visual quality comparison under 2.5 Mbps available bandwidth

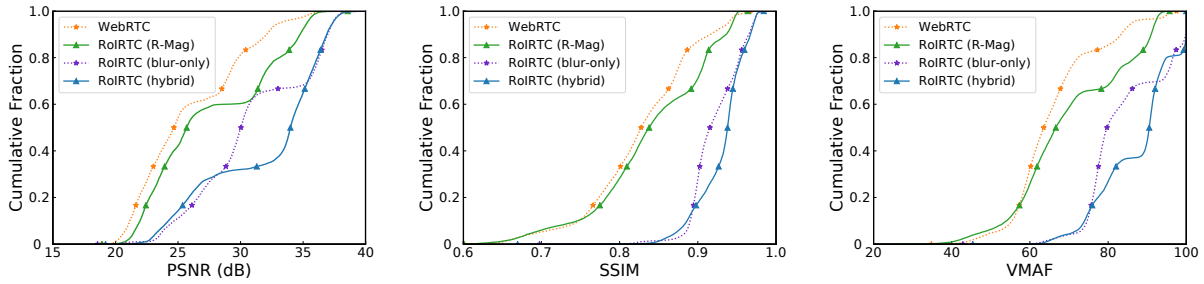


Fig. 4. Overall visual quality comparison under 2.5 Mbps available bandwidth

[2] G.-M.Muntean, G.Ghinea, and T. N.Sheehan, "Region of interest-based adaptive multimedia streaming scheme," *IEEE Transactions on Broadcasting*, vol. 54, no. 2, pp. 296–303, 2008.

[3] R.Guntur and W. T.Ooi, "On tile assignment for region-of-interest video streaming in a wireless lan," in *Proceedings of the 22nd international workshop on Network and Operating System Support for Digital Audio and Video*, 2012, pp. 59–64.

[4] N.Quang Minh Khiem, G.Ravindra, A.Carlier, and W. T.Ooi, "Supporting zoomable video streams with dynamic region-of-interest cropping," in *Proceedings of the first annual ACM SIGMM conference on Multimedia systems*, 2010, pp. 259–270.

[5] A.Mavlankar, P.Agrawal, D.Pang, S.Halawa, N.-M.Cheung, and B.Girod, "An interactive region-of-interest video streaming system for online lecture viewing," in *2010 18th International packet video workshop*. IEEE, 2010, pp. 64–71.

[6] "Emphasis MAP in NVENCODE API ," https://docs.nvidia.com/video-technologies/video-codec-sdk/nvenc-video-encoder-api-prog-guide/index.html#emphasis-map.

[7] M.Lee, W.Park, S.Lee, and S.Lee, "Distracting moments in videoconferencing: A look back at the pandemic period," in *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, 2022.

[8] E. B.Goldstein and L.Cacciamani, *Sensation and perception*, Cengage Learning, 2021.

[9] D.Roberts, T.Duckworth, C.Moore, R.Wolff, and J.O'Hare, "Comparing the end to end latency of an immersive collaborative environment and a video conference," in *2009 13th IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications*. IEEE, 2009, pp. 89–94.

[10] J.Ryoo, K.Yun, D.Samaras, S. R.Das, and G.Zelinsky, "Design and evaluation of a foveated video streaming service for commodity client devices," in *Proceedings of the 7th International Conference on Multimedia Systems*, 2016, pp. 1–11.

[11] D.Pang, S.Halawa, N.-M.Cheung, and B.Girod, "Mobile interactive region-of-interest video streaming with crowd-driven prefetching," in *Proceedings of the 2011 international ACM workshop on Interactive multimedia on mobile and portable devices*, 2011, pp. 7–12.

[12] L.Xie, X.Zhang, and Z.Guo, "Cls: A cross-user learning based system for improving qoe in 360-degree video adaptive streaming," in *Proceedings of the 26th ACM international conference on Multimedia*, 2018, pp. 564–572.

[13] O.Eltobgy, O.Arafa, and M.Hefeeda, "Mobile streaming of live 360-degree videos," *IEEE Transactions on Multimedia*, vol. 22, no. 12, pp. 3139–3152, 2020.

[14] H.Schulzrinne, S.Casner, R.Frederick, and V.Jacobson, "Rtp: A transport protocol for real-time applications," Tech. Rep., 2003.

[15] R.Droste, J.Jiao, and J. A.Noble, "Unified image and video saliency modeling," in *European Conference on Computer Vision*. Springer, 2020, pp. 419–435.

[16] "Toward A Practical Perceptual Video Quality Metric," https://medium.com/netflix-techblog/toward-a-practical-perceptual-video-quality-metric-653f208b9652.