**Option 2: Implement gaze-aware live VR streaming based on HLS, DASH or WebRTC**

VR content such as 360◦ video is typically provided with high definition (e.g., 4K, 8K). The vision of users in the virtual environment evolves in a sphere, which unfolds 360◦ horizontally and 180◦ vertically. Streaming HD video consumes a lot of bandwidth. On the other hand, users only view a part of the spherical data called the **Field of View** (FoV). Humans can see about 220◦ horizontally of FoV naturally, while modern consumer-grade HMDs have a FoV within 90◦ and 120◦ horizontally and vertically. It would be a waste of bandwidth to transmit ultra-high resolution video in other fields than the currently concentrated FoV. We call it **gaze-aware VR streaming**. From the user perspective, the VR content should be delivered within a **low latency** (typically less than 100ms), and the **resolution should be as high as possible**. From a network perspective, the bandwidth cost should be as low as possible. The whole VR streaming solution should be **scalable**, concerning the potential large number of users.

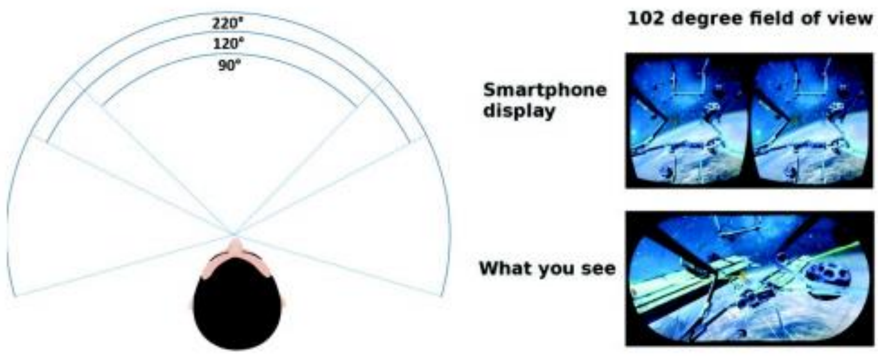


Figure 1. Field of View (FoV) using head mounted display.

In this assignment, you are expected to design and implement gaze-aware **live** VR streaming based on an existing streaming protocol. You can for example choose to build your solution on top of HLS, DASH or WebRTC. Regarding the demo, you can download a 360 VR video from the Internet (e.g. https://www.mettle.com/360vr-master-series-free-360-downloads-page/) , and use it for testing.

For testing, you can implement a simple client in one of the following approaches:

- A Unity or Unreal Engine based VR app that supports gaze tracking and live VR streaming. The app can run on devices like HTC Vive Eye. (Consider this option if you already have experience with Unity or Unreal Engine)
- Implement a simple Android/iOS application. It simulates the head movement by rotating the mobile device, and uses the motion sensors (e.g., gyroscope, accelerometer) embedded on the mobile devices to detect the movement. The application informs the server the movement, and playbacks the received 360 VR video.
- Implement a simple GUI client on your computer and use keyboard or mouse to control the changes in FoV. The client can playback the received 360 VR video.

You need to demo the client and server to our course assistants. Regarding evaluation, you need to report the latency since the movement is detected until the view is updated on the client, throughput, achieved resolution under different network conditions (e.g., high speed, low speed networks). In addition, you need to analyze the scalability of your solution (i.e., test the performance when the number of clients increases).

We recommend you complete the assignment following the steps below.

**Step 1:** Discuss within the group the use cases. Note that use cases define how users would interact with the app. You can use the following template to describe use cases. You should be able to clearly define entities and their roles in the system after this step.

| |
|---|
| *Use case name:*<br>*Actors involved: (e.g. user, client, streaming server and etc.)*<br>*Preconditions:*<br>*Steps:*<br>        *The user xxxx*<br>        *xxx*<br>        *xxx*<br>*Error:*<br>    *e.g. If xxx does not xxx, xxx will be xxx*<br>*Post-conditions:*<br>      *e.g. The client displays xxx* |

**Step 2:** Go through the use cases and summarize the functional requirements. Note that use cases do not typically describe non-functional requirements and constraints. You can identify non-functional requirements by analyzing the operational environment (e.g., potentially large number of users, bandwidth constraint, etc.), the expectations from user experience perspective (e.g., low latency, high resolution), and any other relevant concerns. At this step, you can read some papers about gaze-aware VR streaming, for example, the ones listed in the references.

**Step 3:** Conduct a technical survey to decide which streaming protocol to use for building the solution. In practice, you can compare the designs of HLS, DASH and WebRTC, and run some simple tests as well. Keep in mind that you may need to make some changes into the selected protocol to implement all the functional requirements. When more than one protocol can fulfill the functional requirements, you can check which one could better fulfill the non-functional requirements and would require less modifications to the selected protocol.

**Checkpoint (15.2.2023):** You can submit a summary of requirement analysis and the initial design of the solution via MyCourses. Also present it in a peer review session on 16.2.2023 for example. Feedback on the design will be given then to each group.

**Step 4:** After selecting the streaming protocol (remember to be able to explain your selection criteria later in the report), you can create a detailed design of each functionality. Note that you may need a different protocol for sending the FoV or motion information from the receiver to the sender. It is not acceptable to directly copy a design from the literature or any existing open-source projects.

Here are some questions to think about at Step 3 and Step 4.

- Which data-sharing architecture to follow, distributed or centralized one?

- How to inform the sender of the currently concentrated FoV? Which protocol to use?
- How to segment a frame in the way that each segment corresponds to a FoV?
- How to choose a different bitrate for each segment during encoding?
- How to adapt the bitrate to changing network conditions? Is this supported automatically by the existing streaming protocols?
- Does it require mapping from equirectangular to gnomonic projection [3] and vice versa? (Hint: 360 degree video applications have historically resorted to first mapping spherical pixels to a rectangular surface, then encoding these rectangular images. There are also new ways presented in the literature, e.g. [4])
- Is the solution scalable?

**Step 5:** Implement the solution (both client and server). Remember to divide programming tasks among group members.

**Step 6:** Test the solution with 1- N clients under different network conditions, and check if all the functional and non-functional requirements have been satisfied. If you are wondering which metrics to use for evaluation, you can read for example [1] and [5].

**Step 7:** Summarize the design and experimental results in your final report. You should try to also compare your design with previous ones (e.g. [1], [2], [5]), and discuss the limitations of your own design.

**References:**
[1] P. Lungaro, R. Sjöberg, A. J. F. Valero, A. Mittal and K. Tollmar, "Gaze-Aware Streaming Solutions for the Next Generation of Mobile VR Experiences," in *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 4, pp. 1535-1544, April 2018, doi: 10.1109/TVCG.2018.2794119.
[2] M. Hosseini, "View-aware tile-based adaptations in 360 virtual reality video streaming," 2017 IEEE Virtual Reality (VR), Los Angeles, CA, 2017, pp. 423-424, doi: 10.1109/VR.2017.7892357.
[3] Equirectangular-toolbox. https://github.com/NitishMutha/equirectangular-toolbox
[4] Chao Zhou, Zhenhua Li, and Yao Liu. 2017. A Measurement Study of Oculus 360 Degree Video Streaming. In Proceedings of the 8th ACM on Multimedia Systems Conference (MMSys'17). Association for Computing Machinery, New York, NY, USA, 27–37. DOI:https://doi.org/10.1145/3083187.3083190
[5] Shibo Wang, Shusen Yang, Hailiang Li, Xiaodan Zhang, Chen Zhou, Chenren Xu, Feng Qian, Nanbin Wang, and Zongben Xu. 2022. SalientVR: saliency-driven mobile 360-degree video streaming with gaze information. In Proceedings of the 28th Annual International Conference on Mobile Computing And Networking (MobiCom '22). Association for Computing Machinery, New York, NY, USA, 542–555. https://doi.org/10.1145/3495243.3517018

**Deliverables:**

- **Interim report (deadline: 15.2.2023).** It is enough to submit a Powerpoint presentation. The idea is to get feedback from the teacher and course assistants.
- Present your **demo** to our course assistant and submit its source code on MyCourses by **17.3.2023**. Instead of a live demo, you can also choose to create a demo video and just show the demo video.
- **Final group report (deadline: 21.3.2023).** You can find a template on the next page.

- **Description of Team work** (0.5 pages, deadline: 21.3.2023). This should be included in a final report of Part 3.
  - Describe how you have worked as a team (e.g. regular meetings, workshops, and etc.)
  - State clearly the responsibilities of each team member. (e.g. literature survey, programming tasks, network measurement, report writing, and etc.)
  - Do you think the workload and responsibilities have been equally shared?
- **An individual essay** (0.5 - 1 page, deadline: 21.3.2023) on what you have learnt from the assignments, what you think are the most challenging parts, and what the good things and bad things about teamwork are. Feel free to discuss other topics as well.

***Template of the final report:***

*List each group member's name and student number*

*Chapter 1: System architecture (1-2 pages)*

*Draw a figure that illustrates the high level architecture of the system. Add a brief description of the figure.*
*List the functional requirements and non-functional requirements*

*Chapter 2: Streaming Protocol (2-3 pages)*
*Which streaming protocol do you choose? Why?*
*Any changes made to the existing protocols?*
*How to implement the gaze-aware VR streaming based on the selected protocols?*

*Chapter 3: Evaluation ( 2-4 pages)*
*Experimental setup*
*Test cases (#client, network conditions)*
*Result analysis (e.g. latency, resolution)*
*Does your design fulfill all the requirements?*

*Chapter 4: Discussion (1 page)*
*Discuss the differences compared with previous designs (in the literature or open source projects)*
*Discuss the limitations of your own design*

## Assessment Matrix (max 50 points + 2 bonus points)

### Interim report submitted at checkpoint 1 (max 5 points):

| Topic (weight) | Unacceptable (0) | Marginal (1) | Acceptable (2) | Exceptional (3) |
|---|---|---|---|---|
| **Requirement analysis (1)** | Little or no grasp of the problem. | Some understanding of the problem. | Overall sound understanding of the problem.<br>Use cases are clearly defined. | A list of functional and non-functional requirements is provided. |
| **Project management (1)** | Next steps are not planned. | Next steps are clearly defined. | Next steps are clearly defined. The schedule and task distribution are well planned. | |

### Demo (max 18 points)

| Topic (weight) | Marginal (1) | Acceptable (2) |
|---|---|---|
| **Inform the sender about the currently concentrated FoV (3)** | The receiver can obtain the information about the currently concentrated FoV. | The receiver can send the information about the currently concentrated FoV to the sender. |
| **Encoding/decoding (3)** | Divide a frame into segments and encode them with different visual quality (e.g., bitrate, resolution, etc.). | Divide a frame into segments and encode them with different visual quality.<br><br>The visual quality of the segments can change according to the FoV |
| **Streaming (3)** | The client can playback the viewing part (FoV) of the 360 video correctly. The viewing part changes along with the change of FoV. | Support automatic adaptation of resolutions to changing network conditions. |

### Final report (max 25 points)

| Topic (weight) | Marginal (1) | Acceptable (2) | Exceptional (3) |
|---|---|---|---|
| **Requirement analysis (1)** | Some understanding of the problem. | Overall sound understanding of the problem.<br><br>Functional and non-functional requirements are clearly defined. | |
| **Technical survey (1)** | A literature survey on streaming protocol is done. | Some experiments have been done to compare different streaming protocols. | |
| **Design (3)** | The design can fulfill most of the functional and non-functional requirements. | The design can fulfill the functional and non-functional requirements. | Additionally, detailed design, including changes into the existing protocols, is provided. State machine, message sequence chart and/or class |

| | | | |
|---|---|---|---|
| | | The selection criteria of the streaming protocol to use are explained. | diagram are used properly for explaining the design. |
| **Evaluation (3)** | Measure the latency, resolution, and throughput when there is 1 to N clients. Use tables or figures to present the results. | Run the experiment under changing network conditions and use the experimental results to prove that the solution can adapt resolutions to the network conditions. | Run the experiment under changing network conditions and use the experimental results to prove that the solution can adapt resolutions to the network conditions.<br><br>Analyze the scalability of the solution.<br><br>Discuss the limitations of the proposed solution, and compare your own design with previous works. |
| **Academic writing (1)** | The report is otherwise easy to follow, but some important details are missing. | The report is easy to follow and the ideas are well expressed. The paper is well written, except for a few places which require clarification. | It is well written and concise. Ideas are well expressed. The report is well organized and easy to follow. Plots and diagrams are readily understandable and they support the text. |

## Individual Essay (max 2 points)

| **Topic (weight)** | **Marginal (1)** | **Acceptable (2)** |
|---|---|---|
| **Reflection (1)** | Cover at least one of the following perspectives:<br>- What have you learnt in these assignments?<br>- What are the most challenging parts in these assignments?<br>- What do I think are the good and bad things about teamwork? | Cover all the following perspectives:<br>- What have you learnt in these group assignments?<br>- What are the most challenging parts in this process?<br>- What do I think are the good and bad things about teamwork? |

## Team work (max 2 bonus points)

| **Topic (weight)** | **Unacceptable (0)** | **Acceptable (1)** | **Exceptional (2)** |
|---|---|---|---|
| Team work (1) | Tasks are not assigned to each member. The group does not manage to complete the tasks together. | Responsibility and workload is equally distributed between group members. It is fair to everybody. | Everybody shares his/her knowledge, eagerly working for achievement of the learning goal. Sharing the tasks is fair, and that helps achieve the learning goal. |