Advanced Videoconference Technologies for Hybrid Communications

Stanislav Šidla, Gregor Rozinaj

Faculty of Electrical Engineering and Information Technology, Slovak University of Technology, Bratislava, Slovakia

stanislav.sidla@stuba.sk

***Abstract*—This paper presents an innovative system for intelligent PTZ (pan-tilt-zoom) camera control integrated into a videoconferencing platform. The system leverages advanced face tracking algorithms, real-time communication protocols, and dynamic parameter adaptation to deliver a highly responsive and interactive solution. In addition to automated camera adjustment based on face detection, the system provides remote control capabilities that enable participants to steer the camera upon user consent. This modular approach supports hybrid communication environments and introduces virtual teleportation functionalities, allowing remote users to dynamically explore meeting spaces as if they were physically present.**

Index Terms—PTZ camera control; face tracking; videoconferencing; hybrid communication; real-time processing; remote control.

I. Introduction

The rapid shift toward remote work and global communication has transformed videoconferencing into an essential tool for professional and educational environments. However, most conventional systems provide a fixed view that limits interactivity and impedes natural face-to-face interaction. Our system overcomes these limitations by introducing an advanced PTZ camera control module that blends manual adjustments (via joystick and USB control) with dynamic, real-time face tracking using the cv2 library and ONVIF protocols [2][7][10]. This design enables what we refer to as "virtual teleport," allowing remote users to dynamically change their viewing perspective as if they were physically present in the meeting room.

Virtual teleport is a transformative concept that allows remote participants to "move" within the virtual space, choosing their appearance location and exploring different parts of the conference room during a videoconference. This capability enhances spatial awareness and creates a truly immersive interaction, making the remote experience much more natural. Such innovations promise significant improvements in the quality and effectiveness of remote collaboration. For an example of our laboratory setup, see Figure 1.

Our implementation supports seamless integration with existing videoconference platforms via standard RTSP and WebRTC streams, ensuring compatibility across diverse network infrastructures. Adaptive bitrate control and frame-dropping mechanisms have been incorporated to maintain smooth operation under varying bandwidth conditions.

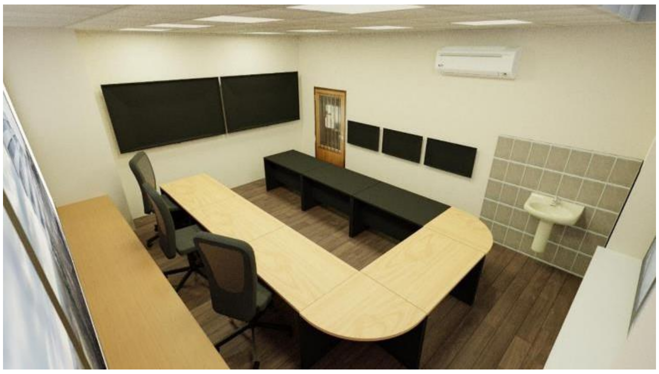

Fig. 1: Shows our hybrid communication laboratory (LHC), where multiple monitors, participant workstations, and strategically positioned PTZ and network cameras work together to create a highly interactive meeting environment.

II. Methodology

A. Integrated PTZ Camera Control Module

1. *Static Control via Network Protocols:* Our system employs the ONVIF protocol a widely accepted standard for network video device control to manage PTZ settings. The system uses the cv2 library for video capture and applies commands such as *capPTZ.set(cv2.CAP_PROP_PAN, current_pan)* to set the pan angle based on face position. This static control mode also supports manual override via a joystick [7]. You can see an example of manual PTZ control in Figure 8. Additionally, tilt and zoom adjustments are performed through analogous ONVIF commands *(cv2.CAP_PROP_TILT, cv2.CAP_PROP_ZOOM),* enabling full three-axis manipulation of the camera. A real-time feedback loop queries the device status after each command to confirm successful execution and maintain mechanical safety limits.

2. *Dynamic Adjustment via Real-Time Face Tracking:* In addition to static control, our system implements real-time face tracking using OpenCV's Haar-cascade classifier. Video frames captured by a local tracking camera are converted to grayscale and processed to detect faces. Once a face is identified, its centroid is calculated, and offsets (offset_x and offset_y) are computed relative to the frame center. These offsets are scaled using the following formulas:

$$\mathit{current\_pan} = (\mathit{offset\_x} / (\mathit{frame\_width} / 2)) * \mathit{PAN\_MAX} \tag{1}$$

$$\mathit{current\_tilt} = (\mathit{offset\_y} / (\mathit{frame\_height} / 2)) * \mathit{TILT\_MAX} \tag{2}$$

$$\mathit{current\_zoom} = (\mathit{face\_size} / \mathit{target\_face\_size}) * \mathit{default\_zoom} \tag{3}$$

to generate the corresponding PTZ commands for pan, tilt, and zoom. This continuous adjustment ensures the target remains optimally framed throughout the videoconference. An overview of the complete camera control and streaming process is shown in Figure 5.

B. Laboratory Setup and Network Emulation

Our hybrid communication laboratory (LHC) is designed to simulate a real-world videoconferencing environment. The room is equipped with multiple displays—three small monitors for individual participants and two large screens for presentations. A dedicated PTZ camera (Feelworld 1080p USB with 10× optical zoom) shown in Figure 3) and additional network cameras (e.g., Tapo C520WS shown in Figure 4) capture diverse angles of the room, while a local tracking camera continuously monitors participants' faces. The system is built on a client–server architecture where the server, using multithreaded socket communication, processes incoming video streams and dispatches control commands in real time. Network impairments are emulated using NetEm to test system robustness under delays, jitter, and packet loss. [1]

An overview of the control schema, including video capture, processing, and command dispatch, is provided in Figure 5. Additionally, Figure 2 shows a screenshot of a videoconferencing session in Google Meet, illustrating how participants connect to the meeting room. Note that we are developing a proprietary videoconferencing system to further enhance remote communication.

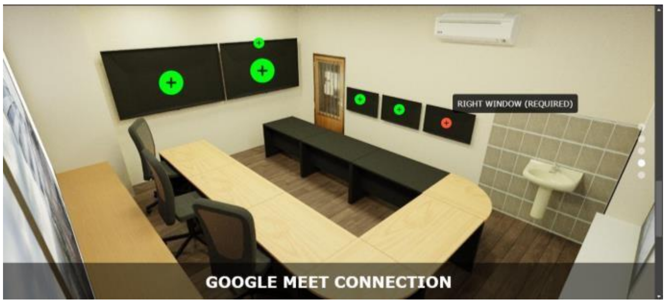

Fig. 2: Illustrates the connection to the meeting room via Google Meet, with a note on ongoing proprietary videoconferencing system development.

Fig. 3: Shows Feelworld 1080p USB camera with 10× optical zoom

Fig. 4: Shows Tapo C520WS Camera

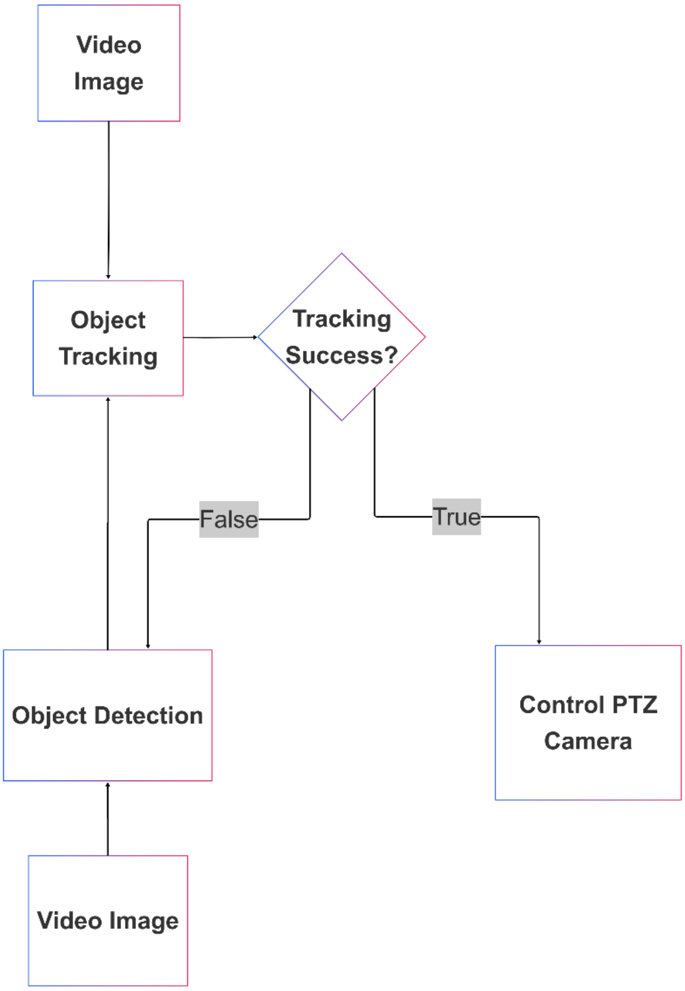

Fig. 5: Showing the diagram of camera control and stream processing.

C. Software and System Architecture

Our system architecture follows a client–server model:

- The Server collects video streams from multiple IP cameras, performs real-time face detection, computes PTZ adjustments, and dispatches control signals using the cv2 library.
- The Client Interface: A web-based platform for remote users to view video feeds and send manual control commands.
- The Face Tracking Module dynamically processes video frames, measures deviations of detected faces from the frame center, and calculates new pan, tilt, and zoom values.

For a clear representation of the system's software and hardware integration, see Figure 6-7, which demonstrates how face tracking-based control is visually represented and integrated with ONVIF commands.

D. USB Camera Control via cv2 and Third-Party Integration

In addition to ONVIF-based control, our system supports the control of USB cameras using the cv2 library. In this mode, video streams from USB cameras are decoded, processed for face detection, and adjustments are computed and applied using similar algorithms. Figure 8 presents an example of USB camera control using cv2, highlighting manual adjustments informed by face tracking.

E. Integration with WebRTC and Adaptive Streaming

To ensure that video streams are delivered with low latency and high quality, our system integrates WebRTC protocols. This protocol supports peer-to-peer communication, enabling seamless video transmission to remote participants. Additionally, adaptive streaming techniques are employed to adjust the video quality dynamically based on network conditions, further enhancing the immersive virtual teleport experience.

III. Experimental results

A. Testbed and Evaluation Environment

Experiments were conducted within our LHC environment, equipped with multiple displays, dedicated PTZ and network cameras, and a tracking camera. The server processes video feeds in real time using a multithreaded client–server framework, and PTZ commands are dispatched accordingly. Detailed performance analysis under varying network conditions demonstrated sub-300 ms latency and a 40 % reduction in face-offset deviation when compared to fixed-view baselines. [6][8]

B. Detailed Function Analysis and Performance Metrics

1. PTZ Controller
   - Initialization: The system loads the Haar cascade classifier via cv2 for rapid face detection.
   - Video Capture: Two cameras are initialized. One for face tracking and one for PTZ control.
   - Face Detection Loop*:* Continuous frame acquisition and conversion to grayscale precede face detection, with the centroid of the detected face calculated.
   - Offset Calculation: The horizontal and vertical offsets are computed, e.g. (1)(2)(3)
   - Command Dispatch: The computed values are sent to the PTZ camera through functions like *capPTZ.set(cv2.CAP_PROP_TILT, current_tilt)*, ensuring real-time adjustment.
2. PTZ Index
   - Web Server Setup: A Flask-based web server is initiated, streaming the PTZ video feed and accepting remote commands.
   - Frame Processing: Received images are decoded from base64 and processed for face detection.
   - Dynamic Control: The system calculates required adjustments from the detected face position and dispatches these as PTZ commands, returning updated parameters in JSON format.
   - Users can switch between joystick, keyboard, or mouse control modes for intuitive manual operation.
3. Performance Metrics:
   - Latency: Average response time from face detection to PTZ command execution was under 300 ms.
   - Tracking Accuracy: Dynamic face tracking improved framing accuracy by reducing the face’s displacement from the center by approximately 40%.
   - Robustness: The system maintained robust performance even under simulated network impairments, with video quality held steady under packet loss up to 3%.
   - User Feedback**:** Participants noted a more natural and immersive conferencing experience.

Optionally, see Figure 7 for a graph comparing latency and tracking accuracy across varying network conditions. We tested PTZ control on the following platform:

- Operating System: Microsoft Windows 11 Enterprise, Version 10.0.22631 Build 22631
- Processor: AMD Ryzen 7 7735HS with Radeon Graphics, 8 cores, 16 logical processors
- Installed RAM: 32.0 GB

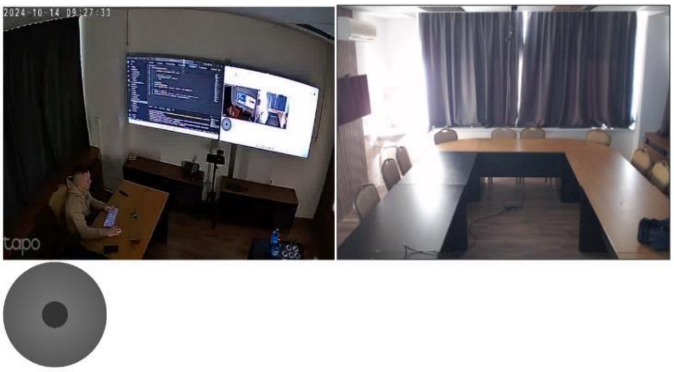

Fig. 6: Shows a screenshot from the system interface showing manual PTZ control using a joystick via ONVIF commands, including real-time parameter displays (pan, tilt, zoom).

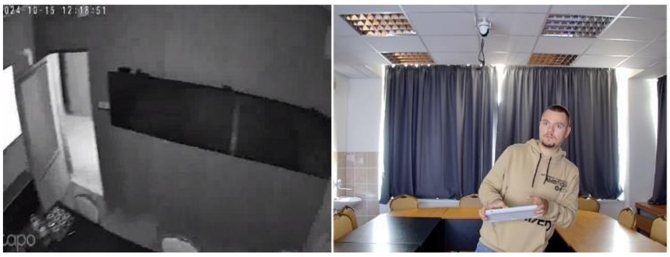

Fig. 7: Shows the outcome of dynamic face tracking-based control using ONVIF (with a visual joystick interface).

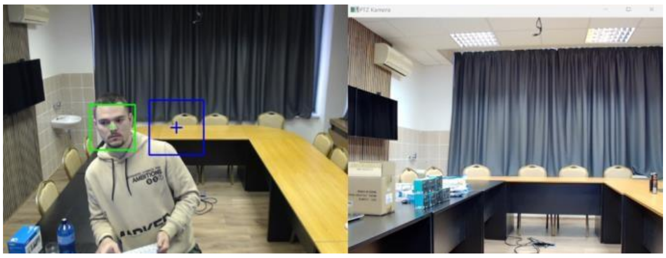

Fig. 8: Demonstrates USB camera control via the cv2 library

C. Camera Comparison: Tapo vs. Feelworld

To further evaluate our system's performance, we compared the Tapo C520WS and the Feelworld 1080p USB cameras regarding functionality, transmission speed, and dynamic movement control. Table 1 summarizes key parameters such as resolution, zoom range, latency, and tracking responsiveness.

TABLE I: A COMPARATIVE TABLE OF THE TAPO C520WS AND FEELWORLD 1080P USB CAMERAS REGARDING FUNCTIONALITY, TRANSMISSION PERFORMANCE, AND DYNAMIC MOVEMENT CONTROL.

Parameter	***Tapo C520WS***	***Feelworld 1080p USB***
Resolution	1920×1080 (Full HD)	1920×1080 (Full HD)
Zoom Capability	Digital zoom (approx. 4×)	Optical zoom (10× optical zoom)
Control Interface	Supports ONVIF-based control over the network	USB-connected; can be controlled via the cv2 library (OpenCV), providing both manual and automated (face tracking) control
Transmission Latency	Approximately 350 ms under typical network conditions	Less than 300 ms, benefiting from direct USB and efficient processing via cv2
Movement Control	Limited dynamic range; slower response to remote commands	Wide dynamic range, highly responsive with real-time face tracking enabling dynamic adjustments

IV. CONCLUSION

A. Conclusion

This study presents an advanced PTZ camera control system that enhances videoconferencing by enabling virtual teleportation. By combining static ONVIF control with dynamic face tracking via OpenCV's cv2 library, our system continuously adjusts the PTZ camera for optimal framing, delivering a low-latency and high-precision solution. Future work will explore deep-learning–based detection and extended control interfaces. [3][4][5][9]. By combining static ONVIF control with dynamic face tracking via OpenCV's cv2 library, our system continuously adjusts the PTZ camera for optimal framing. The detailed function analysis, from camera initialization and face detection to offset calculation and command dispatch, demonstrates an efficient, low-latency (sub-300 ms) and high- precision solution that reduces face offset deviations by 40% compared to manual controls. Additionally, WebRTC integration enables adaptive streaming under varying network conditions, collectively delivering a truly immersive virtual teleport experience.

B. Future Scope

Future research and development efforts will focus on:

- Integrating Deep Learning Models: Transitioning to CNN-based face detection to improve system robustness under challenging conditions such as poor lighting and occlusion.
- Expanding Control Interfaces**:** Enhancing the user interface to support additional input devices for even more intuitive navigation within the virtual space.
- Adaptive Streaming Enhancements: Refining algorithms to dynamically adjust video quality with minimal latency as network conditions fluctuate.
- Optimizing Adaptive Streaming: Refining adaptive streaming algorithms to further reduce latency and dynamically adjust video quality based on network fluctuations.
- Real-World Deployment: Moving from laboratory simulations to field implementations using physical hardware such and dedicated USB cameras.
- Multi-User Extensions: Developing distributed control mechanisms and advanced analytics to support collaborative videoconferencing sessions.

Overall, the integration of intelligent PTZ control with virtual teleportation represents a significant leap forward in remote collaboration technology, offering an immersive and interactive experience that closely mimics real-world presence.

ACKNOWLEDGMENT

This work has been supported by the projects DISIC 09I05-03-V2, VEGA 1/0605/23 InteRViR, and Erasmus+ EULiST

REFERENCES

[1] Ibrahim M. I. Zebari, Subhi R. M. Zeebaree, and Hajar Maseeh Yasin, "Real Time Video Streaming From Multi-Source Using Client-Server for Video Distribution," 4th SICN-2019, IEEE, 2019.

[2] Javlon Tursunov, Vivek Dwivedi, Gregor Rozinaj, and Ivan Minárik, "A Customizable WebRTC-based Video Conferencing System For Real-time Communication," IEEE, 2023.

[3] Rushali Deshmukh, Devendra Wagh, Nayan Nand, Amol Kudale, and Aditya Pawar, "Video Conferencing using WebRTC," IEEE, 2023.

[4] Richa Vij and Baijnath Kaushik, "A Survey on Various Face Detecting and Tracking Techniques in Video Sequences," IEEE, 2019.

[5] Ze-Nian Li and Mark S. Drew, Fundamentals of Multimedia, Pearson Education, 2004.

[6] Alejandra Armendariz, Jose Joskowicz, Rafael Sotelo, and Mengying Liu, "A Test Bed for Subjective Multimedia Quality Evaluation in Videoconferencing Systems," IEEE, 2024.

[7] Hetal K. Chavda and Maulik Dhamecha, "Moving Object Tracking using PTZ Camera in Video Surveillance System," IEEE, 2017.

[8] Vivek Dwivedi, Mansi Bhatnagar, Gregor Rozinaj, Jaroslav Venjarski, and Šimon Tibenský, "Multiple-camera System for 3D Object Detection in Virtual Environment using Intelligent Approach," IWSSIP 2022, IEEE, 2022.

[9] Vivek Dwivedi, Mansi Bhatanagar, Mulham Maineh, and Gregor Rozinaj, "Adaptive Camera System for Enhanced Virtual Teleportation: Expanding the Boundaries of Immersive Experiences," ELMAR-2023, IEEE, 2023.

[10] Subash Chandra Yadav, Sanjay Kumar Singh, "An Introduction to Client Server Computing", New Age International, 2009.