H.265(HEVC):

H.265, also known as High Efficiency Video Coding (HEVC), is a video compression standard that delivers the same high-quality video as its predecessor H.264 but at roughly half the bitrate or file size. Developed by the Joint Collaborative Team on Video Coding (JCT-VC), it was approved as an ITU-T standard in April 2013.

H.264 vs H.265 comparison

HEVC uses larger Coding Tree Units (CTUs) up to 64x64 pixels instead of H.264's smaller 16x16 macroblocks, enabling smarter analysis of uniform areas for better compression.

Developed in 2013 by ITU-T & ISO/IEC MPEG

Successor to H.264/AVC

Designed for efficient high-resolution video delivery

Supports up to **8K UHD.**

Why H.265 Matters

50% bitrate reduction compared to H.264

Enables smooth 4K/8K streaming

Saves bandwidth and storage

Essential for modern broadcasting & streaming platforms

Features	H.264 (AVC)	H.265 (HEVC)
Year Introduced	2003	2013
Compression Efficiency	Standard	~50% better
Bitrate for 1080p	4,500–6,000 kbps	2,250–3,000 kbps
Bitrate for 4K	25–35 Mbps	12–16 Mbps
Successor	H.265	H.266 (VVC)

Applications:

Streaming Services: Netflix, Amazon Prime, YouTube

Broadcasting: UHD TV, satellite transmission

Video Conferencing: High-quality, low-latency calls

Mobile Devices: Efficient recording & playback

Challenges:

Licensing costs (multiple patent pools)

Hardware compatibility issues on older devices

Higher encoding complexity (needs stronger processors)

1) Why HEVC (H.265) helps you

- Keeps video quality high but uses less bandwidth and storage.

- Important for remote control and for logging trails without draining power.

2) What to decide first

- Hardware: onboard encoder or send raw frames to edge computer?

  - Onboard (Jetson, Snapdragon, etc.): lower latency, more compact.

  - Edge: can use stronger encoders, easier updates.

- Resolution and frame rate: pick what your control loop needs.

  - Example: 720p at 15–20 fps is often enough for navigation.

  - If you need detail for rope inspection, use 1080p at 10–15 fps.

- Latency budget: how much delay can your control loop tolerate? 100–200 ms is typical for reactive robotics; aim lower if possible.

3) Simple encoder settings to start with

- Use low-latency mode (avoid long GOPs, fewer B-frames).

- If you can, use hardware encoder (NVENC, Quick Sync, or your SoC's encoder).

- Recommended starting points (adjust by hardware):

  - Preset: veryfast or ultrafast (latency first).

  - Tune: zerolatency (helps reduce buffering).

  - GOP size (keyint): 15–30 (1–2 seconds at 15–20 fps).

  - Bitrate target: start around 2–4 Mbps for 720p at 15–20 fps; adjust up or down based on network.

  - Profile: Main10 for 10-bit if your camera/pipe supports it; otherwise Main.

4) Practical streaming options

- Onboard: encode to HEVC and send via UDP/RTP to edge station or operator.

- Edge: send uncompressed or lightly compressed video to edge PC, encode there, and return decisions quickly.

- Latency-friendly transport: use UDP with small buffers; consider WebRTC for two-way control and streaming if you need bidirectional latency handling.

5) Simple pipeline you can implement now

- Capture: camera at 720p, 15–20 fps.

- Encode: HEVC with low-latency settings (see above).

- Transport: UDP/RTP to edge PC or operator.

- Use: edge PC or onboard decoder for display, and feed video to perception stack (SLAM/obstacle detection).

- Logging: save a lower-bitrate version locally for mission replay.

6) Quick test plan (one afternoon)

- Baseline: 720p, 20 fps, 2 Mbps, on onboard encoder.

  - Measure end-to-end latency (camera to operator display or to edge perception).

  - Check power draw.

- Try lower latency: reduce GOP to 10, 5–10 B-frames if any, see latency drop.

- Test network variations: strong Wi-Fi vs weak signal. Enable a basic ABR if available (switch between 2–4 Mbps).

- Check robot tasks: does video delay affect rope-following accuracy or obstacle avoidance? Note results.

7) What I can tailor for you

- Tell me your exact hardware (camera type, onboard CPU/GPU, edge PC if any).

- Desired resolution/frame rate and max acceptable latency.

- Network type (Wi-Fi, 4G/5G, LOS link) and whether you have a streaming server or ROS 2 setup.