Design and Evaluation of an Affordable and Reliable Pan-Tilt Tracking System

Damian Łoziński
Independent researcher, IEEE Member, Warsaw, Poland

Bartosz Kozłowski
Warsaw University of Technology, Faculty of Electronics and Information Technology, Institute of Control and Computation Engineering, Nowowiejska 15/19, 00-665 Warsaw, Poland

Abstract: This paper presents the design and evaluation of an affordable and reliable pan-tilt tracking system. The proposed solution addresses the limitations of existing professional remote-controlled camera systems by offering a low-cost, modular hardware design combined with a fine-tunable control algorithm for real-time person or face tracking. Leveraging widely available components and 3D printing technology, the system is optimized for ease of production and accessibility. Experimental results demonstrate that the system achieves stable and smooth tracking performance, balancing responsiveness and precision while maintaining affordability. This work contributes to making advanced tracking technologies more accessible for applications such as video production, conferencing, and robotics.

Keywords: pan-tilt control, object tracking, adaptive PID control, face detection, vision systems, low-cost mechatronics, camera automation, embedded systems, signal filtering, 3D printing.

1. Introduction

The increasing demand for dynamic and flexible video production in professional studios necessitates advanced camera systems capable of pan, tilt, and zoom adjustments. Although some professional cameras have begun incorporating automatic tracking features, these systems often lack customization options and frequently require human operators to ensure reliable performance, leading to increased operational costs and reduced accessibility.

This paper addresses the gap in the availability of cost-effective, fine-tunable control algorithms for person or face tracking by introducing an affordable, modular pan-tilt mechanism compatible with user-provided cameras. The solution broadens access to tracking technologies by combining widely available components, the use of 3D printing techniques, and adaptable software.

The research focuses on two primary objectives:

1. Bridging the gap in fine-tunable control algorithms by designing a pan-tilt visual control system that integrates recent advancements in object detection and tracking.
2. Promoting the use of low-cost, high-quality pan-tilt tracking systems by developing a simple, reliable device based on accessible components.

Autor korespondujący:
Damian Łoziński, damian@lozinski.tech

Artykuł recenzowany
nadesłany 29.09.2024 r., przyjęty do druku 17.03.2025 r.

2. Literature Review

2.1. Detectors and Trackers

Object detection and tracking are critical components of any automated pan-tilt camera system, playing a vital role in maintaining target stability and accuracy. Over the past two decades, advancements in both detection and tracking algorithms have significantly improved the performance and feasibility of real-time systems, particularly those operating on low-power hardware.

Object Detection Techniques

Early object detection methods were primarily developed to address computational challenges and were often tailored to specific applications. One of the foundational methods, the *Haar Cascade classifier* introduced in 2001, employed a meta-algorithm approach that showed robust results in image-based object recognition tasks [1, 2]. Subsequently, in 2005, researchers advanced the field with a *Support Vector Machine* (SVM) classifier utilizing *Histograms of Oriented Gradients* (HoGs), demonstrating both speed and accuracy in person detection [3]. These classical methods remain relevant for low-resource environments, such as the Raspberry Pi 3, due to their relatively low computational requirements [41].

In recent years, *Deep Neural Networks* (DNN) have revolutionized object detection by offering highly generalized and accurate models. Prominent DNN-based algorithms include

Fast Recurrent Convolutional Neural Networks (Fast R-CNN) [4, 5], *Single Shot Multibox Detector* (SSD) [6], and the *You Only Look Once* (YOLO) family, including YOLOv1-12, YOLO-pose, YOLO-NAS [7–10]. Additionally, MobileViT [11] and its variants have emerged as efficient vision transformers suitable for resource-constrained environments. These methods enable fast and reliable detection of a broad spectrum of objects, making them suitable for dynamic environments. The proposed system incorporates the YuNet model [12], selected for its compactness, speed, and satisfactory accuracy on low-end computing devices. Additionally, alternative lightweight models such as BlazeFace [13] and MobileFaceNet [14] have shown competitive performance in face detection tasks, particularly in mobile and embedded environments. The integration of models like YOLO-NAS and MobileViT offers further flexibility and efficiency, particularly for embedded and mobile systems. These models are also considered viable for specific hardware constraints. The choice of YuNet aligns with the system's objective of balancing performance, accuracy, and computational efficiency.

Object Tracking Approaches

Object tracking, in contrast to detection, is generally less computationally intensive but equally crucial for maintaining the continuity of target positioning. The *Continuously Adaptive Mean Shift* (CAMSHIFT) algorithm [15] is well-known for its efficiency in short-term tracking, although it may experience drift over longer durations. To address such limitations, correlation filters such as *Minimum Output Sum of Squared Error* (MOSSE) [16] and *Discriminative Correlation Filters* (DCF) [17] have been developed, offering enhanced speed and adaptability. Universal Kalman filters are also widely used in tracking applications, providing predictive modeling that smooths motion and reduces jitter [18, 19]. Kernelized correlation filters that perform efficiently, even under challenging tracking scenarios, were introduced in [20].

Additionally, wavelet transform-based approaches [21] have been explored for feature extraction in combination with particle filters to achieve precise positioning [22]. These techniques are particularly useful when the system needs to predict movement patterns, such as forecasting pan-tilt targeting errors as a time series [23].

Recent advancements in tracking technology focus on integrating detection into the tracking loop, facilitating *Multiple Object Tracking* (MOT) and maintaining unique target identities across frames. Approaches such as *Simple Online and Realtime Tracking* (SORT) [24], OC-SORT [25], Deep OC-SORT [26], DeepSORT [27], ByteTrack [28], and FairMOT [29] exemplify this integration. These methods often leverage state-of-the-art detectors like YOLO or DLA-34 [30], enhancing their tracking performance by providing robust initial detections [42, 43]. Among these, OC-SORT and ByteTrack have shown particular effectiveness in balancing real-time performance and tracking robustness, especially on resource-constrained devices, while Deep OC-SORT extends OC-SORT's capabilities by integrating adaptive re-identification strategies.

Lightweight tracking models, such as those in the OpenCV Zoo[1] project, further demonstrate the trend toward efficiency and speed. NanoTrack [44] and VIT tracker [45] models, for example, are particularly valuable for applications with strict performance constraints, such as unmanned aerial vehicle (UAV) tracking. Moreover, MediaPipe [46], developed by Google, offers a modular and cross-platform solution for face detection, hand tracking, and pose estimation. Its high efficiency and customizability make it a strong candidate for both high-end and low-end devices.

2.2. Pan-Tilt Control

Although significant progress has been made in object detection and tracking techniques, the development of refined control modules for pan-tilt systems is still an area with room for further exploration. Effective pan-tilt control plays an important role in achieving smooth, responsive, and stable camera movements, especially in dynamic environments such as video production studios and industrial monitoring systems.

Existing Approaches to Pan-Tilt Control

A notable contribution to Pan-Tilt-Zoom (PTZ) control systems is presented in [31], where a Haar Cascade classifier is integrated with a Continuously Adaptive Mean Shift (CAMSHIFT) tracker to detect and track faces. The proposed system demonstrated effectiveness in industrial applications; however, it fell short of achieving the smooth camera movements necessary for studio environments. The limitation primarily stemmed from visible iterative movements between panning and zooming, leading to a lack of fluidity in the camera's output.

A more hardware-focused approach is presented in [32], where a servo-motor-controlled pan-tilt visual system was developed. The study revealed that the PID control method delivered a faster transient response compared to the lead-lag approach for both pan and tilt movements. While this research effectively addressed response speed, it also highlighted a broader need for advanced control strategies that maintain object stability within the camera frame without sacrificing movement smoothness.

An innovative method involving automatic tuning of a PID controller for pan-tilt mechanisms using a Neural Network was proposed in [33]. The study aimed to optimize key performance metrics such as overshoot, peak time, and rise time of the PID driver. Although the approach achieved commendable control performance, the emphasis on rapid response introduced a trade-off, with smoothness of camera movement being largely overlooked.

A study on predictive tracking using a pan-tilt camera system [34] demonstrated that employing a *Dynamic Matrix Control* (DMC) strategy, which is a specific implementation of *Model Predictive Control* (MPC), improved tracking performance. The system maintained the target near the center of the image, enhancing recognition and reducing processing time. The DMC approach outperformed traditional PID controllers, highlighting the benefits of predictive strategies in pan-tilt systems.

An alternative control strategy using *Reinforcement Learning* (RL) for tracking distant moving objects was explored in [35]. While this study introduced a novel approach by allowing the entire platform to move, it diverges from the objectives of this research, which focuses on stationary pan-tilt devices. The RL-based method, though effective for mobile platforms, is less applicable in scenarios where mechanical simplicity and static positioning are prioritized.

An adaptive control method for pan-tilt devices is presented in the work of another study [36], where an asymptotically stable adaptive algorithm was proposed. Despite its stability benefits, this approach similarly failed to address the need for smooth and natural camera transitions, which are critical in applications such as live broadcasting and autonomous video production.

2.3. Implications for the Proposed System

Many of the reviewed studies utilize classical object detection and tracking algorithms, relying primarily on techniques such as Haar Cascade classifiers, CAMSHIFT trackers, or basic PID controllers. While these methods have proven effective in certain contexts, their performance may be constrained under complex or fast-changing conditions, particularly in scenarios involving rapid target movement, variable lighting, or temporary occlusions. These limitations highlight the potential benefits of integrating modern deep learning-based detection and tracking models into the control loop.

[1] OpenCV Zoo and Benchmark – models with benchmarks tuned for OpenCV DNN on different platforms https://github.com/opencv/opencv_zoo

The review of existing literature indicates a clear need for a balanced approach to pan-tilt control, one that harmonizes rapid response times with the smoothness of camera movements. Building upon these insights, the proposed system incorporates a hybrid control strategy that combines contemporary detection algorithms with adaptive PID control, predictive filtering, and motion smoothing techniques. This approach aims to deliver a refined pan-tilt control mechanism capable of maintaining target stability while ensuring fluid and professional-grade camera movements.

3. Methodology

This section outlines the rationale, design choices, and architecture of the proposed pan-tilt tracking system. The primary focus is to describe how the system integrates modern detection and tracking algorithms with a simple, reliable mechanical design and an adaptable control strategy. Emphasis is placed on balancing system affordability, ease of production, and performance. The following subsections detail the system's architecture, mechanical components, and control algorithm, providing a comprehensive understanding of how the design choices align with the overarching objectives of the research.

3.1. System Architecture

The system is composed of three primary components that collectively enable real-time object tracking with smooth pan-tilt camera control depicted on fig. 1:

- *Camera* – captures the video feed used for object detection and tracking. The system supports flexible camera integration, allowing users to attach their preferred camera model. The video stream is processed in real-time by the software module to maintain continuous focus on the target.
- *Software Module* – responsible for executing the detection, tracking, and control algorithms. It runs on a general-purpose processing unit such as a personal computer or embedded system. The software analyzes the incoming video stream, detects and tracks the target, and computes appropriate commands to steer the pan-tilt mechanism.
- *Pan-Tilt Device hardware* – the electro-mechanical component of the system, consisting of a two-axis rotation mechanism driven by stepper motors and a custom-designed control board. It is designed to accommodate a wide variety of cameras via a standard photographic mount. The hardware receives control commands from the software and executes precise pan and tilt movements via the dedicated motor interface.

3.2. Mechanical Design

The mechanical subsystem consists of a compact two-axis assembly enabling independent pan and tilt motion. A standardized photographic mount ensures broad compatibility with various camera models and streamlines both production and integration.

The design emphasizes mechanical simplicity and reliability by minimizing the number of components and reducing the camera's distance to the rotation axes, thereby improving stability and tracking accuracy. Additive manufacturing is used for fabrication, supporting modularity and cost-effective replication suitable for research, education, and automated video applications.

The mechanism utilizes stepper motors (28BYJ-48) controlled via an Arduino Micro and ULN2003 drivers, forming a microcontroller-based control module. All components are selected for availability, low power consumption, and ease of assembly. The modular, 3D-printed structure is compatible with standard tripod fittings and is designed for straightforward maintenance and adaptation to various use cases.

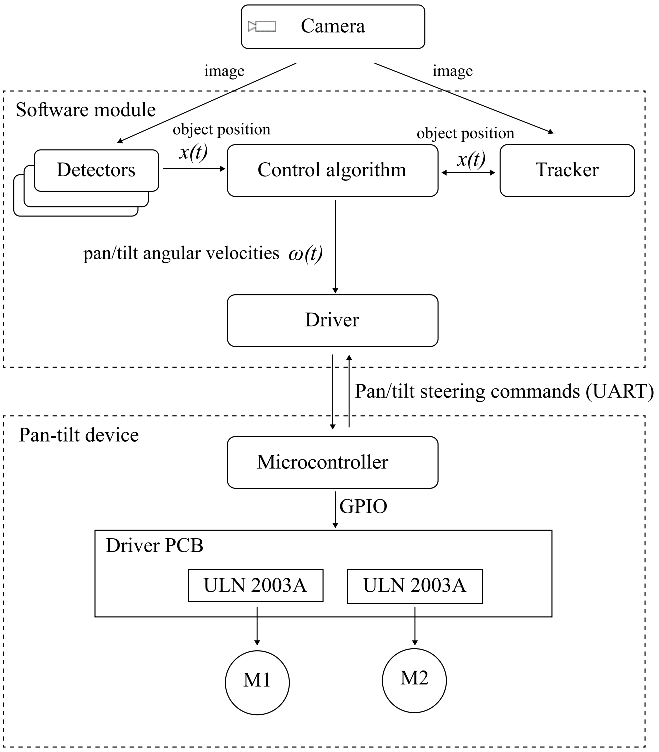


Fig. 1. High-level architecture of the pan-tilt tracking system. The diagram shows data flow between the video input, detection/tracking/control software, and actuator hardware
Rys. 1. Architektura wysokopoziomowa systemu śledzenia typu pan-tilt. Schemat przedstawia przepływ danych pomiędzy sygnałem wideo, modułami detekcji, śledzenia i sterowania, a układem wykonawczym

3.3. Control Algorithm and Software Design

The software and control algorithm of the pan-tilt tracking system are designed to provide efficient object detection, tracking, and smooth control of the camera's movement. The system operates in a closed-loop manner, continuously adjusting the pan-tilt mechanism to maintain the target object at the center of the frame.

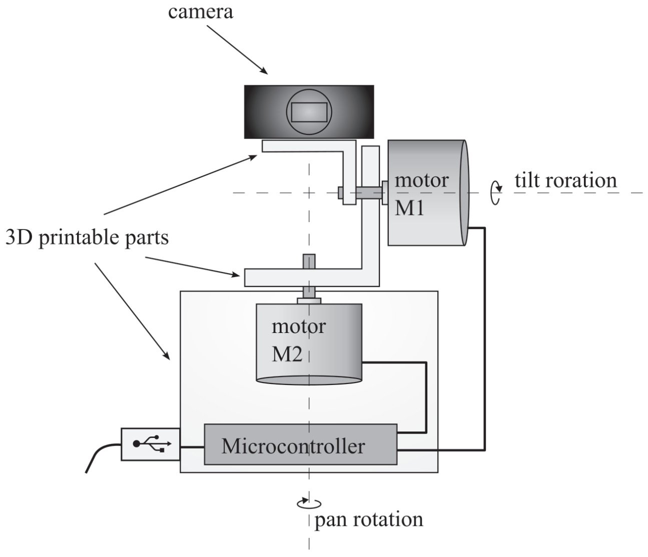


Fig. 2. Device mechanical design. A front view of the device, showing motor placement and rotation axes, with M1 representing the tilt axis and M2 the pan axis
Rys. 2. Konstrukcja mechaniczna urządzenia. Widok z przodu ukazujący rozmieszczenie silników oraz osie obrotu, gdzie M1 odpowiada za oś pochylenia (tilt), a M2 za oś obrotu (pan)

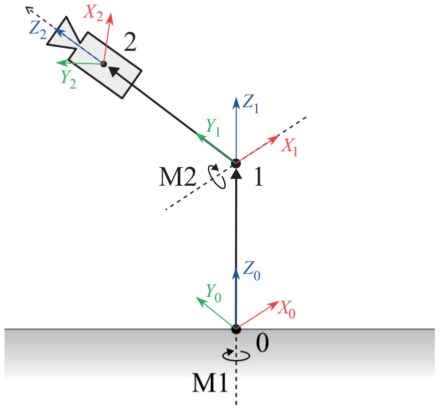


Fig. 3. Coordinate frames applied to the pan-tilt mechanism. Frame 0 is fixed to the base, Frame 1 rotates about the vertical axis (pan), and Frame 2 rotates about the horizontal axis (tilt), attached to the camera mount

Rys. 3. Układy współrzędnych zastosowane w mechanizmie pan-tilt. Układ 0 jest trwale związany z podstawą, układ 1 realizuje rotację wokół osi pionowej (pan), natomiast układ 2 — wokół osi poziomej (tilt) i jest zamocowany do uchwytu kamery

Detection-Tracking-Control Pipeline

The system is organized into a structured pipeline consisting of detection, tracking, and control modules, working in a closed loop. The *detector* identifies objects of interest within video frames, the *tracker* maintains object continuity across frames, and the *controller* computes corrective movements to keep the target centered.

Key software components:

– *Detectors*: Identify objects in each frame, operating in stateless (frame-by-frame) or stateful (tracking-assisted) modes.
– *Trackers*: Maintain object positions across frames, compensating for occlusions or detection losses.
– *Controller*: Calculates appropriate pan-tilt movements based on tracker data.
– *Driver*: Executes motor commands based on control inputs, ensuring smooth and precise movements.

This modular setup supports integration of various detection and tracking algorithms while maintaining stable and responsive control.

Closed-Loop Control Strategy

The system uses a closed-loop control mechanism that continuously adjusts the pan and tilt angles to keep the tracked object centered in the camera's field of view. The control error, denoted as ε, is defined as the vector difference between the object's current position x and the center of the image plane c, i.e., $\varepsilon = c - x$ (fig. 4). Both x and c refer to 2D coordinates on the camera image plane, not to positions in 3D space. The coordinate system is normalized such that the origin (0, 0) is located at the center of the image, with the bottom-left corner having coordinates (−1, −1) and the top-right corner at (1, 1). This error serves as input to the control algorithm.

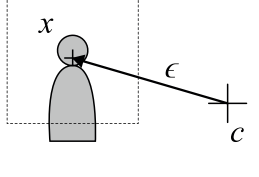


Fig. 4. Camera image with denoted error ε, i.e., vector connecting origin of the image plane *c* with the center *x* of *a* detected object

Rys. 4. Obraz z kamery z zaznaczonym wektorem błędu ε, definiowanym jako różnica między środkiem płaszczyzny obrazu c a pozycją x wykrytego obiektu

To achieve stable tracking, the control system adjusts the pan-tilt velocities in real time, ensuring smooth and continuous movement. The system supports two primary operational modes:

– **Fast tracking mode**: Prioritizes rapid response to object motion, keeping the target centered with minimal delay. This mode is ideal for robotics applications where immediate interaction with the object is necessary.
– **Smooth tracking mode**: Emphasizes gradual movements to enhance video quality, avoiding abrupt changes in camera direction. This mode is suited for video production, conferences, and live streaming applications.

The system can be tuned to function in either of these modes or in a hybrid configuration that balances responsiveness and smoothness.

PID and Adaptive Control Strategies

The pan-tilt mechanism utilizes a PID (Proportional-Integral-Derivative) controller to regulate its motion. The PID controller processes the positional error ε and computes the required angular velocity:

$$PID(\varepsilon) = K_p \cdot \varepsilon + K_i \cdot \int \varepsilon dt + K_d \cdot \frac{d\varepsilon}{dt}$$

– K_p (Proportional Gain): Determines the response to the current error ε. A higher K_p results in a faster response but may cause overshooting.
– K_i (Integral Gain): Accounts for the accumulation of past errors by integrating ε over time. This helps eliminate steady-state errors but can introduce lag if set too high.
– K_d (Derivative Gain): Reacts to the rate of change of the error $d\varepsilon/dt$. It helps dampen oscillations and improves stability by predicting future error trends.

For dynamic adjustments, the system employs an *adaptive PID control* mechanism in which the controller gains K_p, K_i, and K_d are continuously updated based on the object's motion dynamics. Let $|\dot{\varepsilon}(t)|$ denote the velocity and $|\ddot{\varepsilon}(t)|$ the acceleration of the tracking error. The gains are computed as:

$$\hat{K}_p(t) = K_{p0} + \alpha_p |\dot{\varepsilon}(t)| + \beta_p |\ddot{\varepsilon}(t)|$$

$$\hat{K}_d(t) = K_{d0} + \alpha_d |\ddot{\varepsilon}(t)|$$

$$\hat{K}_i(t) = K_{i0} e^{-\gamma_i |\dot{\varepsilon}(t)|}$$

where K_{p0}, K_{d0}, K_{i0} are baseline (initial) gain values and α_p, β_p, α_d, and γ_i are empirically tuned coefficients. Higher velocities and accelerations increase the proportional and derivative gains, improving response time during rapid movements. Simultaneously, the integral gain is attenuated to mitigate windup.

To ensure continuity and avoid abrupt parameter shifts, gain values are updated using exponential smoothing:

$$K_{\square}(t) = gK_{\square}(t-1) + (1-g)\hat{K}_{\square}(t), \quad 0 < g < 1, \quad \forall \square \in \{p, d, i\}$$

This approach enables the controller to adjust its behavior based on target dynamics, maintaining stable operation across varying conditions.

The computed angular velocity is further smoothed using the following equation:

$$\omega_t = s\omega_{t-1} + (1-s)PID(\varepsilon)$$

where ω_t is the new angular velocity and $0 \le s < 1$ is the smoothing factor.

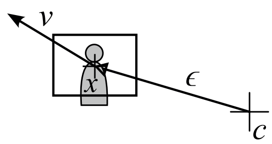


Fig. 5. Dynamic tolerance frame during rapid movements of the target; tolerance threshold shrinks; *x*, *v* – position and velocity of the tracking target in the camera image plane, *ϵ* – tracking error vector, *c* – origin of the camera image plane

Rys. 5. Dynamiczne okno tolerancji podczas szybkiego ruchu obiektu. Próg tolerancji ulega rozszerzeniu w celu zwiększenia responsywności układu. Symbolami x i v oznaczono odpowiednio pozycję oraz prędkość śledzonego obiektu, ε reprezentuje wektor błędu śledzenia, a c odnosi się do środka płaszczyzny obrazu kamery

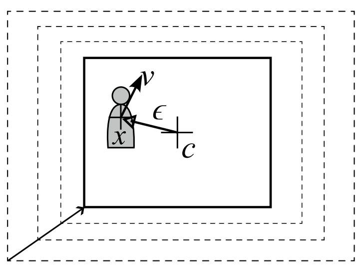


Fig. 6. Dynamic tolerance frame when the target remains steady. Tolerance threshold initially expands, then starts shrinking

Rys. 6. Dynamiczne okno tolerancji w przypadku ustabilizowanej pozycji obiektu. Próg tolerancji początkowo rośnie, następnie zostaje zawężony w celu zwiększenia dokładności pozycjonowania

To further enhance stability, the system integrates **Dynamic Tolerance Frame** (**DTF**) filtering, which dynamically adjusts the tracking threshold based on the object's motion state. When the tracked object is in motion (i.e., $|\varepsilon|$ or $d\varepsilon/dt$ is large), the tolerance window **contracts immediately** (fig. 5), enabling precise and responsive corrections. Once the object stabilizes near the center of the frame (fig. 6), the system **temporarily expands** the tolerance to suppress minor jitter (e.g., head movements), preventing unnecessary camera oscillations. Over time, this window **gradually contracts** again to refine centering accuracy. This mechanism incorporates *temporal hysteresis* – the system does not react instantly to small changes but instead waits to confirm stability before narrowing the tolerance, ensuring both smoothness and precision.

These operational phases are depicted in fig. 7, where the system transitions between the *React*, *Buffer*, and *Refine* states depending on the magnitude and dynamics of the tracking error.

Low-Pass Filtering and Predictive Control

To mitigate tracking noise and improve movement smoothness, the system incorporates low-pass filtering techniques. A heatmap-based filter (fig. 8) is applied to stabilize position estimates and suppress erratic movements. The image is divided into a grid whose number of columns and rows is configurable via system parameters. Each cell in the grid represents a spatial segment that accumulates 'heat' when an object is detected within it or its vicinity. The heating intensity decays geometrically with distance from the object's center, according to a configurable radius parameter. Similarly, a cooling factor controls the rate at which cells lose heat over time when no object is present nearby. All parameters – including grid resolution, heating decay factor, cooling rate, and heating radius – are tunable, allowing for system calibration depending on the application context. This dynamic accumulation and decay mechanism helps identify consistently active regions and improves the stability and confidence of the tracking system.

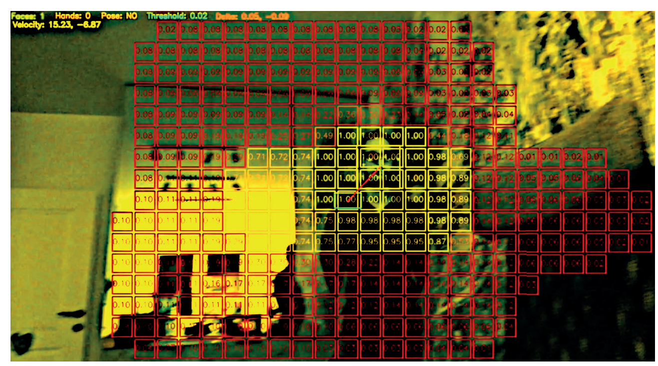


Fig. 8. Heatmap-based low-pass filtering enhances the probability of target detection in the vicinity of the current target, effectively suppressing false positives in unrelated regions

Rys. 8. Filtracja dolnoprzepustowa oparta na mapie ciepła zwiększa wiarygodność lokalizacji celu w jego bezpośrednim sąsiedztwie, jednocześnie skutecznie eliminując fałszywe detekcje w pozostałych obszarach obrazu

Additionally, the system leverages predictive control to anticipate object trajectories. Using a third-order Taylor expansion, the system estimates future positions based on current velocity and acceleration (fig. 9):

$$x(t+dt) \approx x(t) + v(t)dt + 0.5a(t)dt^2$$

where $x(t)$ is the object's current position in the viewport plane, $v(t)$ is velocity on that plane, and $a(t)$ is acceleration. This predictive approach reduces latency in tracking and minimizes overshoot when following fast-moving objects.

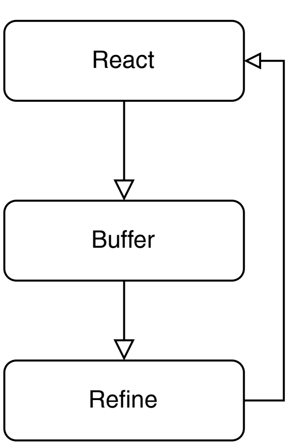


Fig. 7. State transitions in the Dynamic Tolerance Frame (DTF) control logic. In the *React state*, large tracking error causes the tolerance window to contract immediately for reactive response. Upon centering, the system enters the *Buffer state*, during which the window temporarily expands to prevent jitter caused by small involuntary motions. Over time, the system transitions to the *Refine* state, where the window gradually narrows again to refine alignment. This process reflects a form of temporal hysteresis, balancing responsiveness with precision

Rys. 7. Diagram stanów w logice sterowania Dynamicznego Okna Tolerancji (DTF). W stanie React duży błąd śledzenia skutkuje natychmiastowym zawężeniem okna tolerancji. Po ustabilizowaniu obiektu system przechodzi do stanu Buffer, w którym okno tymczasowo się rozszerza w celu redukcji drgań wywołanych drobnymi, mimowolnymi ruchami. Następnie układ przechodzi do stanu Refine, w którym tolerancja jest stopniowo zawężana, co sprzyja precyzyjnemu centrowaniu. Proces ten odzwierciedla zjawisko histerezy czasowej, równoważąc szybkość reakcji z dokładnością

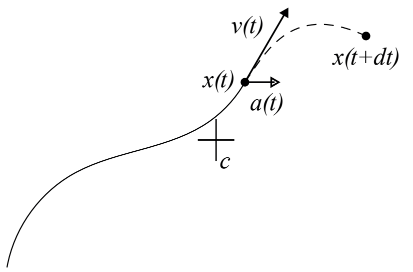


Fig. 9. Trajectory of the object in the camera image plane and an attempt to predict its position at time *t* + *dt* based on its instantaneous velocity *v*(*t*) and acceleration vector *a*(*t*)
Rys. 9. Trajektoria obiektu w płaszczyźnie obrazu kamery wraz z przewidywanym położeniem w chwili *t* + *dt*, wyznaczonym na podstawie jego chwilowej prędkości *v*(*t*) oraz przyspieszenia *a*(*t*) zgodnie z rozwinięciem Taylora rzędu trzeciego

Communication and Execution Flow

The software communicates with the pan-tilt hardware via UART/USB, transmitting control commands in real-time. The driver module processes steering instructions and executes motor commands through a queue system, ensuring smooth and non-blocking operation.

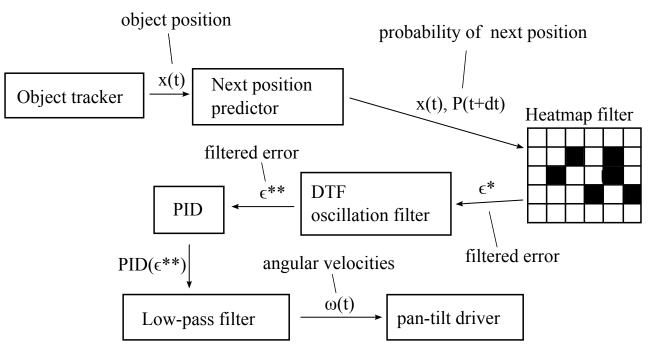


Fig. 10. Flow diagram of the pan-tilt control pipeline; ϵ^* represents the error filtered by the heatmap, while ϵ^{**} denotes the error further processed by the DTF filter. Each filter either forwards the error value or nullifies it, depending on the current state
Rys. 10. Schemat przepływu danych w łańcuchu sterowania pan-tilt. Wektor ε* oznacza błąd przefiltrowany przy użyciu mapy ciepła, natomiast ε** reprezentuje wartość dodatkowo przetworzoną przez filtr DTF. Każdy z filtrów, w zależności od aktualnego stanu układu, przekazuje wartość błędu do kolejnego etapu przetwarzania lub ją wygasza, zapewniając stabilność i płynność działania systemu

The final execution flow (fig. 10) of the control system follows these stages:
1. *Detection*: Identifies objects in video frames.
2. *Tracking*: Maintains object continuity across frames.
3. *Error Computation*: Calculates deviation from the center of the frame.
4. *Predictive Analysis*: Estimates future positions based on motion trends.
5. *Oscillation filters*: Uses heat map to filter rapid and big oscillations and DTF to filter small oscillations of higher frequency.
6. *PID and Adaptive Control*: Computes motor velocity adjustments.
7. *Filtering and Stability Enhancements*: Applies low-pass filters.
8. *Motor Execution*: Sends velocity commands to the pan-tilt mechanism.

This structured approach ensures precise, stable, and adaptable tracking performance, making the system suitable for diverse applications ranging from robotics to automated video production.

4. Implementation

The pan-tilt tracking system is implemented through the integration of software and hardware subsystems designed to operate in real-time and maintain tracking stability. This section provides a detailed overview of the system's technical architecture, including the selection of software frameworks, hardware control logic, and communication protocols.

4.1. Software Implementation

The software is implemented in Python, utilizing established computer vision and control libraries to ensure modularity and real-time performance.

Python was selected primarily for its extensive ecosystem of libraries and the efficiency it offers during the prototyping phase. While the language is interpreted and inherently high-level, its widespread adoption in scientific computing stems from its ability to interface with low-level, compiled numerical libraries such as BLAS and LAPACK through frameworks like NumPy.

In this system, critical numerical computations – including matrix operations, filtering, and control law evaluations – are executed outside the Python interpreter, leveraging architecture-optimized binaries. This approach ensures that performance-critical tasks benefit from native execution speed, while high-level control flow remains accessible and adaptable within Python.

Software Stack and Components

The software stack consists of modular components for detection, tracking, control, and communication. It is optimized for embedded and general-purpose systems, with a focus on efficiency, real-time performance, and ease of integration.

Detection and Tracking:
- *YuNet* [12]: Lightweight face detector optimized for embedded systems, based on a compact convolutional neural network architecture that allows real-time inference on devices with limited computational power. It supports face alignment and bounding box regression
- *OC-SORT* [25]: An efficient tracking algorithm that extends Simple Online and Realtime Tracking (SORT) by incorporating appearance information and motion consistency, improving robustness in occlusions and crowded scenes. It utilizes Kalman filters and association metrics to maintain consistent object identities across frames.

Control:
- *Adaptive PID Control*: Ensures smooth and responsive pan-tilt motion; parameters adjust based on target dynamics.

Filtering and Prediction:
- Low-pass filters, heatmap stabilization, and Taylor-based position prediction reduce jitter and latency.

Communication:
- *pySerial*: Handles serial communication (UART) over USB between the host system and the Arduino microcontroller. It enables real-time command transmission with configurable parameters such as baud rate, parity, and stop bits, ensuring low-latency and reliable control signaling [47].

Frameworks and Libraries:
- *OpenCV*: The core library for image processing and computer vision tasks. It supports frame capture, object detection, contour analysis, and geometric transformations, which are essential for implementing real-time visual tracking pipelines [37].
- *NumPy*: A fundamental package for numerical computation in Python, used for efficient array operations. It is utilized in control logic for PID computation, low-pass filtering, and predictive trajectory modeling [38].

Communication Between Components

The communication subsystem enables low-latency and deterministic data exchange between the host processor and actuator hardware. The host analyzes the video stream in real time, computes the required angular velocities for pan and tilt, and transmits them over a UART interface via USB.

An Arduino-based microcontroller receives these commands through a dedicated interface and generates step signals to maintain the desired angular velocities on both axes. This separation of responsibilities allows for modular design, with the host handling high-level vision processing and the microcontroller managing low-level motor control.

Control commands are issued asynchronously, at a variable frequency that adapts automatically to the system state – e.g., tracking error or target motion dynamics. This decoupling from the camera's frame rate ensures smooth actuation and responsiveness even under processing delays or dropped frames.

4.2. System Integration and Testing

The integration process involved testing various tracking scenarios, optimizing *Adaptive PID* control parameters, and fine-tuning object detection and tracking settings. The system was validated through a series of experiments in different environments, ensuring:

- *Real-time performance*: The tracking system operates with minimal latency.
- *Stability and robustness*: Smooth camera movements with minimal oscillations.
- *Adaptability*: Effective tracking under varying lighting conditions and target speeds.

The implemented system successfully combines Python-based tracking algorithms with Arduino-driven motor control, providing a cost-effective and reliable solution for automated pan-tilt tracking applications.

5. Experimental Setup and Evaluation

To rigorously assess the performance of the proposed pan-tilt tracking system, a structured experimental framework was developed. The objective was to evaluate the system's accuracy, responsiveness, and robustness under a variety of real-world conditions. Experiments were conducted in an indoor environment designed to simulate diverse operational scenarios, including variations in illumination, object motion profiles, and temporary visual occlusions. The evaluation aimed to quantify the system's operational limits and identify critical performance determinants.

5.1. Experimental Setup

The system under evaluation consisted of a custom-built pan-tilt platform actuated by stepper motors, with real-time control managed by an Arduino Micro microcontroller. The control software, including detection and tracking algorithms, was executed on a general-purpose computing platform equipped with an Intel Core i7 processor and 16 GB of RAM. The visual input was captured by a camera operating at a resolution of 1080p and a frame rate of 30 frames per second, providing adequate spatial and temporal resolution for robust tracking.

Experimental conditions were configured to reflect typical usage scenarios. Lighting conditions were varied systematically to include high illumination levels (above 1000 lux), standard indoor lighting (~300 lux), and low-light settings. Motion scenarios were divided into three categories based on target velocity: slow (approximately 1 m/s), moderate (up to 3 m/s), and rapid motion characterized by abrupt direction changes. In each case, the target maintained a nominal distance of 3 meters from the camera.

To evaluate occlusion resilience, the target was periodically obstructed by other objects, simulating real-world interruptions. Recovery performance, including reacquisition time and accuracy post-occlusion, was closely monitored. The overall experimental design emphasizes repeatability and relevance to real deployment conditions.

5.2. System Performance

Evaluation Metrics

The performance of the pan-tilt tracking system was assessed using a set of standard quantitative metrics frequently cited in PTZ tracking literature [36, 39, 40]. These metrics enable objective comparison with previous research and provide a multidimensional evaluation of system behavior:

- *Center Location Error* (CLE) – represents the Euclidean distance between the ground-truth and predicted target centers in image coordinates. It measures the system's ability to align with the actual target trajectory:

  $$CLE_t = \left\| C_{gt}(t) - C_{pred}(t) \right\|,$$

  where $C_{gt}(t)$ is the ground-truth center of the target and $C_{pred}(t)$ is the predicted center at time t;

- *Overlap Ratio* (OVR) – defined as the intersection-over-union (IoU) between predicted and ground-truth bounding boxes. It indicates spatial consistency of target localization:

  $$IoU_t = \frac{A_{gt}(t) \cap A_{pred}(t)}{A_{gt}(t) \cup A_{pred}(t)},$$

  where $A_{gt}(t)$ and $A_{pred}(t)$ are the ground-truth and predicted bounding box areas;

- *Target-to-Center Error* (TCE) – measures the distance between the center of the frame and the predicted target location. It is a key metric for assessing control precision:

  $$TCE_t = \left\| C_{frame} - C_{pred}(t) \right\|,$$

  where C_{frame} is the center of the frame and $C_{pred}(t)$ is the predicted target position;

- *Track Fragmentation Rate* (TFR) – quantifies the proportion of frames in which the target leaves the camera's field of view, indicating continuity and robustness:

  $$TFR = \frac{N_{lost}}{N_{total}} \times 100\%,$$

  where N_{lost} is the number of lost frames and N_{total} is the total number of frames;

- *Response Time* (RT) – time elapsed between a target's sudden movement and the corresponding system adjustment. It reflects real-time responsiveness:

  $$RT = t_{adjust} - t_{event},$$

  where t_{event} is the time of the sudden motion and t_{adjust} is the time the system reacted;

- *Motion Smoothness Index* (MSI) – captures variability in angular velocity to assess motion fluidity:

  $$MSI = \frac{1}{T} \sum_{t=1}^{T} (\omega_t - \bar{\omega})^2,$$

  where ω_t is the angular velocity at time t and $\bar{\omega}$ is the average angular velocity.

Performance Summary

The quantitative results from the experimental trials are summarized below:

Metric	Value	Observations
CLE	~31 px	Degrades under high-speed target motion
OVR	0.78	Slightly reduced during erratic motion
TCE	< 10 % of frame width	Brief excursions from center observed
TFR	15 %	Primarily due to occlusion and abrupt movements
RT	135 ms	Delay increases marginally under complex dynamics
MSI	3.1 deg^2/s^2 (adaptive), 12.4 deg^2/s^2 (static PID)	Adaptive control yields smoother tracking (75 % reduction)

5.3. System Robustness Analysis

Lighting Conditions

Under reduced lighting (300 lux), tracking accuracy degraded by approximately 10 %, and the *Track Fragmentation Rate* (TFR) increased by 5 %. The system maintained functional performance in low-light conditions down to 200 lux, though with elevated jitter and slower reacquisition times.

Target Velocity

Reliable tracking was achieved for object speeds up to approximately 2.5 m/s at a 3-meter distance from the camera. Performance degradation was observed beyond 3.4 m/s, attributed primarily to mechanical inertia and step resolution limitations of the pan-tilt system.

Occlusion Resilience

The system successfully reacquired the target in 75 % of occlusion scenarios within 400 milliseconds. Reacquisition accuracy diminished in the presence of cluttered backgrounds or prolonged target absence. False positive detections were more frequent when occlusions exceeded 1.2 seconds, suggesting a limitation of the current tracking pipeline under extended visual interruptions.

6. Interpretation and Limitations

The results obtained from experimental evaluation confirm that the proposed system maintains robust tracking capabilities under standard operating conditions. It exhibits reliable performance in moderate lighting, target velocities up to 2.5 m/s, and transient occlusions shorter than 1.2 seconds. The integration of adaptive control strategies, predictive filtering, and low-pass smoothing enables the system to balance responsiveness with motion stability.

However, certain limitations were observed. Under rapid target motion exceeding 3.4 m/s or during prolonged occlusions, the tracking accuracy and reacquisition reliability were reduced. These issues are primarily attributable to mechanical constraints (e.g., limited torque and resolution of the stepper motors) and the limitations of the detection-tracking pipeline in handling complex or ambiguous visual input.

Additionally, performance under suboptimal lighting (below 200 lux) revealed increased latency and oscillatory behavior, indicating the sensitivity of the current vision pipeline to illumination quality. Further, the system lacks native multi-object tracking and re-identification capabilities, which constrains its applicability in more complex scenarios.

These limitations point to several directions for future improvement, including the use of high-torque actuators, integration of inertial sensors or depth cameras, and the development of advanced tracking algorithms incorporating temporal consistency and context-aware re-identification mechanisms.

7. Conclusion and Future Work

This study presents the development and empirical validation of a cost-effective and adaptable pan-tilt tracking system. The proposed architecture integrates modular hardware, adaptive control strategies, and lightweight vision algorithms to deliver real-time object tracking with a balance between precision, responsiveness, and implementation cost.

Experimental results confirm the system's ability to maintain stable tracking across a variety of operational conditions, including moderate occlusions, dynamic target velocities, and changes in lighting. The design prioritizes modularity, ease of replication, and compatibility with open-source software and off-the-shelf hardware.

Despite its robust performance, the system exhibits limitations when subject to high-speed target motion, prolonged occlusions, or low-light scenarios. These findings identify clear directions for improvement, such as enhancing actuation capabilities, increasing optical and sensory redundancy, and applying data-driven control schemes.

Future development efforts will focus on extending multi-target support, integrating semantic re-identification techniques, and evaluating deployment feasibility on embedded platforms. Additional research will address hardware constraints through the use of higher-precision motors and energy-efficient designs, thereby broadening the system's applicability across domains such as autonomous robotics, live event production, and surveillance.

References

1. Viola P., Jones M., *Rapid object detection using a boosted cascade of simple features*, Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001, DOI: 10.1109/CVPR.2001.990517.
2. Lienhart R., Maydt J., *An extended set of Haar-like features for rapid object detection*, Proceedings of the International Conference on Image processing, IEEE, 2002, DOI: 10.1109/ICIP.2002.1038171.
3. Dalal N., *Histograms of oriented gradients for human detection*, Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, DOI: 10.1109/CVPR.2005.177.
4. Girschick R., *Fast R-CNN*, Proceedings of the IEEE International Conference on Computer Vision, 2015, 1440–1448, DOI: 10.1109/ICCV.2015.169.
5. Ren S., He K., Girshick R., Sun J., *Faster R-CNN: Towards real-time object detection with region proposal networks*, Advances in Neural Information Processing Systems, 2015, 91–99.

6. Liu W., Anguelov D., Erhan D., Szegedy C., Reed S., Fu C.-Y., Berg A.C., *SSD: Single Shot MultiBox Detector*, European Conference on Computer Vision, Springer, LNIP Vol. 9905, 2016, 21–37.
7. Hussain M., *YOLO-v1 to YOLO-v8, the rise of YOLO and its complementary nature toward digital manufacturing and industrial defect detection*, "Machines", Vol. 11, No. 7, 2023, DOI: 10.3390/machines11070677.
8. Tian Y., Ye Q., Doermann D., *YOLOv12: Attention-Centric Real-Time Object Detectors*, "Computer Vision and Pattern Recognition", 2025, DOI: 10.48550/arXiv.2502.12524.
9. Terven J., Córdova-Esparza D.-M., Romero-González J.-A., *A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS*, "Machine Learning and Knowledge Extraction", Vol. 5, No. 4, 2023, 1680–1716, DOI: 10.3390/make5040083.
10. Dong C., Du G., *An enhanced real-time human pose estimation method based on modified YOLOv8 framework*, "Scientific Reports", Vol. 14, 2024, DOI: 10.1038/s41598-024-58146-z.
11. Wadekar S.N., Chaurasia A., *MobileViTv3: Mobile-friendly vision transformer with simple and effective fusion of local, global and input features*, "Computer Vision and Pattern Recognition", 2022, DOI: 10.48550/arXiv.2209.15159.
12. Wu W., Peng H., Yu S., *YuNet: A tiny millisecond-level face detector*, "Machine Intelligence Research", Vol. 20, No. 5, 2023, 656–665, DOI: 10.1007/s11633-023-1423-y.
13. Bazarevsky V., Kartynnik Y., Vakunov A., Raveendran K., Grundmann M., *BlazeFace: Sub-millisecond neural face detection on mobile GPUs*, "Computer Vision and Pattern Recognition", 2019, DOI: 10.48550/arXiv.1907.05047.
14. Chen S., Yang Y., Liu W., Shao J., Yan J., *MobileFaceNets: Efficient CNNs for accurate real-time face verification on mobile devices*, Springer, LNIP Vol. 10996, 2018, 428–438, DOI: 10.1007/978-3-319-97909-0_46.
15. Bradski G.R., *Real time face and object tracking as a component of a perceptual user interface*, Proceedings Fourth IEEE Workshop on Applications of Computer Vision (WACV'98), 1998, 214–219, DOI: 10.1109/ACV.1998.732882.
16. Bolme D., Beveridge J., Draper B., Lui Y., *Visual object tracking using adaptive correlation filters*, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010, 2544–2550, DOI: 10.1109/CVPR.2010.5539960.
17. LukežičA., Vojíř T., Čehovin L., Matas J., Kristan M., *Discriminative correlation filter with channel and spatial reliability*, "International Journal of Computer Vision", Vol. 126, 2018, 671–688, DOI: 10.1007/s11263-017-1061-3.
18. Tsai C.Y., Song K.T., Dutoit X., Brussel H., Nuttin M., *Robust mobile robot visual tracking control system using self-tuning Kalman filter*, Proceedings of the 2007 IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA), 2007, 161–166, DOI: 10.1109/CIRA.2007.382860.
19. Chen X., Wang X., Xuan J., *Tracking multiple moving objects using unscented Kalman filtering techniques*, "Computer Vision and Pattern Recognition", 2018, DOI: 10.48550/arXiv.1802.01235.
20. Henriques J., Caseiro R., Martins P., Batista J., *High-speed tracking with kernelized correlation filters*, "IEEE Transactions on Pattern Analysis and Machine Intelligence", Vol. 37, No. 3, 2015, 583–596, DOI: 10.1109/TPAMI.2014.2345390.
21. He C., Zheng Y.F., Ahalt S.C., *Object tracking using the Gabor wavelet transform and the golden section algorithm*, "IEEE Transactions on Multimedia", Vol. 4, No. 4, 2002, 528–538, DOI: 10.1109/TMM.2002.806534.
22. Rui T., Zhang Q., Zhou Y., Xing J., *Object tracking using particle filter in the wavelet subspace*, "Neurocomputing", Vol. 119, 2013, 125–130, DOI: 10.1016/j.neucom.2012.03.036.
23. Kozlowski B., *Wavelet enhanced analytical and evolutionary approaches to time series forecasting*, Adaptive and natural computing algorithms, ICANNGA 2007, LNTCS Vol. 4431, 2007, 49–57, DOI: 10.1007/978-3-540-71618-1_6.
24. Bewley A., Ge Z., Ott L., Ramos F., Upcroft B., *Simple online and realtime tracking*, "Computer Vision and Pattern Recognition", 2016, DOI: 10.1109/ICIP.2016.7533003.
25. Cao J., Pang J., Weng X., Khirodkar R., Kitani K., *Observation-centric SORT: Rethinking SORT for robust multi-object tracking*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, 9686–9696, DOI: 10.1109/CVPR52729.2023.00934.
26. Maggiolino G., Ahmad A., Cao J., Kitani K., *Deep OC-SORT: Multi-pedestrian tracking by adaptive re-identification*, "Computer Vision and Pattern Recognition", 2023, DOI: 10.48550/arXiv.2302.11813.
27. Wojke N., Bewley A., Paulus D., *Simple online and realtime tracking with a deep association metric*, 2017 IEEE International Conference on Image Processing (ICIP), 2017, 3645–3649, DOI: 10.1109/ICIP.2017.8296962.
28. Zhang Y., et al., *ByteTrack: Multi-object tracking by associating every detection box*, "Computer Vision and Pattern Recognition", 2022, DOI: 10.48550/arXiv.2110.06864.
29. Zhang Y., Wang C., Wang X., Zeng W., Liu W., *FairMOT: On the fairness of detection and re-identification in multiple object tracking*, "Computer Vision and Pattern Recognition", 2021, DOI: 10.1007/s11263-021-01513-4.
30. Dai Y., Liu W., Li H., Liu L., *Efficient foreign object detection between PSDs and metro doors via deep neural networks*, "IEEE Access", Vol. 8, 2020, 46723–46734, DOI: 10.1109/ACCESS.2020.2978912.
31. Mian A., *Realtime face detection and tracking using a single Pan, Tilt, Zoom camera*, 23rd International Conference Image and Vision Computing New Zealand, 2008, DOI: 10.1109/IVCNZ.2008.4762103.
32. Yosafat R., Machbub C., Hidayat E.M.I., *Design and implementation of Pan-Tilt control for face tracking*, 7th IEEE international conference on system engineering and technology (ICSET), 2017, 217–222, DOI: 10.1109/ICSEngT.2017.8123449.
33. Permatasari D., Maharani D.A., *Backpropagation neural network for tuning PID pan-tilt face tracking*, 3rd International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE), 2018, 357–361, DOI: 10.1109/ICITISEE.2018.8720968.
34. Nebeluk R., Zarzycki K., Seredyński D., Chaber P., Figat M., Domański P.D., Zieliński C., *Predictive tracking of an object by a pan-tilt camera of a robot*, "Nonlinear Dynamics", Vol. 111, No. 9, 2023, 8383–8395, DOI: 10.1007/s11071-023-08295-z.
35. Yang J., Tang Z., Pei Z., Song X., *A novel motion-intelligence-based control algorithm for object tracking by controlling PAN-tilt automatically*, "Mathematical Problems in Engineering", Vol. 2019, No. 1, 2019, DOI: 10.1155/2019/9602460.
36. Petrov P., *An adaptive pan-tilt camera control for visual target tracking*, 29th National Conference with International Participation (TELECOM), 2021, 125–128, DOI: 10.1109/TELECOM53156.2021.9659665.

37. Bradski G., *The OpenCV library*, "Dr. Dobb's Journal of Software Tools", Vol. 25, No. 11, 2000, 120, 122–125.
38. Harris C.R. et al., *Array programming with NumPy*, "Nature", Vol. 585, No. 7825, 2020, 357–362, 2020, DOI: 10.1038/s41586-020-2649-2.
39. Li X., Xu H., Wang Y., *Reproducible evaluation of pan-tilt--zoom tracking*, "IEEE Transactions on Circuits and Systems for Video Technology", Vol. 29, No. 12, 2019, 3609–3622, DOI: 10.1109/TCSVT.2018.2873151.
40. Kimura H., Kimura H., Umeda S., *High speed and wide range object tracking system using pan-tilt cameras*, in 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2006, 1019–1024, DOI: 10.1109/IROS.2006.281669.

Other sources

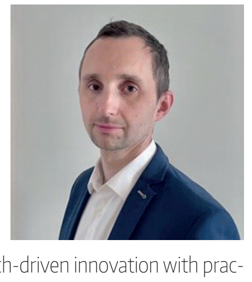
41. Raspberry Pi 3 Model B+, 2024. [www.raspberrypi.com/products/raspberry-pi-3-model-b].
42. *Understanding multiple object tracking using DeepSORT*, Learn OpenCV, 2022, [https://learnopencv.com/understandingmultiple-object-tracking-using-deepsort].
43. Valechha L., *Object tracking and reidentification with FairMOT*, Learn OpenCV, 2022, [https://learnopencv.com/object-tracking-andreidentification-with-fairmot].
44. Chu H., *NanoTrack*, 2022, [https://github.com/HonglinChu/NanoTrack].
45. Pengyu L., *GSoC 2023 realtime object tracking models*, 2023, [https://github.com/opencv/opencv_zoo/tree/main/models/object_tracking_vittrack].
46. Google, MediaPipe, 2024, [https://mediapipe.dev].
47. Liechti C., *pySerial: Python serial port access library*, 2020, [https://pyserial.readthedocs.io/].

Projekt i ocena niskokosztowego i niezawodnego systemu śledzenia z dwuosiowym mechanizmem obrotu (pan-tilt)

Streszczenie: W artykule przedstawiono projekt oraz ocenę niskokosztowego i niezawodnego systemu śledzenia z dwuosiowym mechanizmem obrotu (pan-tilt). Proponowane rozwiązanie stanowi odpowiedź na ograniczenia występujące w profesjonalnych, zdalnie sterowanych systemach kamer, oferując modułową konstrukcję sprzętową oraz precyzyjnie dostrajalny algorytm sterowania, umożliwiający śledzenie osób lub twarzy w czasie rzeczywistym. System wykorzystuje ogólnodostępne komponenty oraz technologię druku 3D, co sprzyja łatwości jego wytwarzania i szerokiej dostępności. Wyniki badań eksperymentalnych potwierdzają, że zaprojektowane rozwiązanie zapewnia stabilne i płynne działanie, skutecznie równoważąc responsywność z precyzją, przy zachowaniu niskich kosztów implementacji. Przedstawiony system stanowi istotny krok w kierunku upowszechnienia zaawansowanych technologii śledzenia w takich obszarach zastosowań, jak produkcja wideo, wideokonferencje czy robotyka.

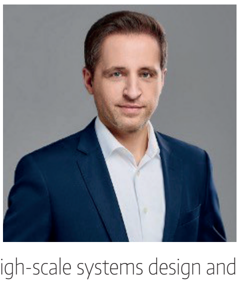
Słowa kluczowe: sterowanie pan-tilt, śledzenie obiektów, adaptacyjne sterowanie PID, wykrywanie twarzy, systemy wizyjne, mechatronika niskokosztowa, automatyzacja kamer, systemy wbudowane, filtracja sygnałów, druk 3D.

Damian Łoziński, MSc
damian@lozinski.tech
ORCID: 0000-0002-9702-576X



Damian Łoziński is an independent researcher and analytical technologist specializing in AI/ML, robotics, and visual tracking systems. He holds a Master's degree in Applied Mathematics from the University of Warsaw. With over a decade of experience, he has developed solutions in computer vision, object tracking, data analytics, and control systems. He is also the founder of multiple technology ventures, including a process mining platform, combining research-driven innovation with practical software and hardware implementation.

Bartosz Kozłowski, PhD Eng.
bartosz.kozlowski@pw.edu.pl
ORCID: 0000-0001-7804-2735



Bartosz Kozlowski is an Adjunct Professor at the Institute of Control and Computation Engineering, Warsaw University of Technology. He received his PhD and MSc degrees in Computer Science from the same university. Professionally, he has been active in the technology industry, leading diverse and globally distributed engineering teams, building technological capabilities and talent capacities in international organizations. His scientific interests include high-scale systems design and software engineering, artificial intelligence and machine learning, operations research (including multi-criteria decision making), and their applications to automation in decision support systems, forecasting, and signal analysis (including wavelet-based methods), as well as aspects of embedded and hardware-integrated solutions, especially for IoT.