2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Pose Retargeting from a Single RGB Camera: Optimization-Based Hand Pose Retargeting and Wrist Pose Estimation

Longrui Chen[1], Lipeng Chen[2,†], Kunpeng Yao[1], Mehmet Dogar[1]

***Abstract*— Robot teleoperation plays a crucial role in collecting data for large-scale imitation learning. Inferring operator's hand pose is crucial for vision-based teleoperation, and current solutions either rely on additional neural network training or hardware to infer the operator's wrist pose. To our knowledge, there is no open-source, general teleoperation toolkit that can be easily deployed to retarget both hand and wrist poses from a single RGB camera. In this paper, we propose OAT (Optimization-based hAnd pose retargeting and wrisT pose estimation), a streamlined approach to retarget human hand and wrist pose to the robot. We leverage the off-the-shelf MediaPipe framework to estimate the operator's hand pose and employ an optimization-based method to infer the operator's wrist pose within the camera frame by 2D/3D hand joint matching. This integrated pipeline facilitates teleoperation from virtually any location using any device equipped with an RGB camera, offering a highly accessible and easily implementable solution. Furthermore, a hand-based camera calibration optimization is proposed to improve the accuracy of wrist pose estimation. In addition to minimal hardware requirements and deployment convenience, our system also demonstrates superior real-time performance compared to state-of-the-art vision-based teleoperation methods.**

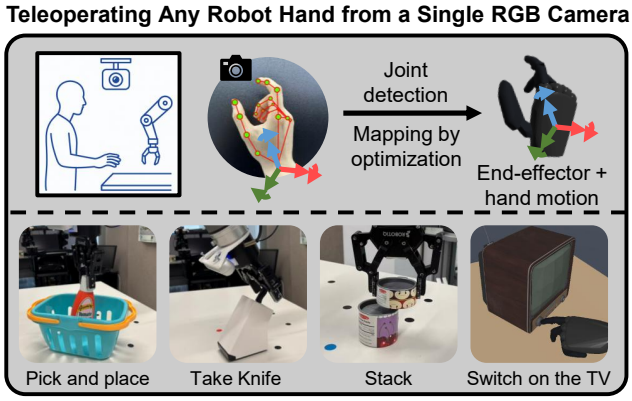

Fig. 1: An overview of OAT, a system for hand pose retargeting and wrist pose estimation using a single RGB camera. Four scenarios are presented to show the capability of OAT, with three on a real robot, and one dexterous teleoperation in simulation. OAT eliminates the need for depth cameras, special devices, or additional network training.

I. INTRODUCTION

Teleoperation is vital for remote robot control [1] and collecting demonstrations in imitation learning [2]. Existing methods fall into three categories: simple wrist control [1], [3], device-assisted teleoperation [4]–[9], and vision-based systems [10]–[14].

Simple interfaces, such as keyboards, joysticks, and spacemouce, are low-cost and easy to implement but offer limited degrees of freedom and unnatural control schemes, often hindering complex manipulation tasks. Specialized devices, such as VR headsets, sensor gloves, and handheld controllers, improve fidelity and user embodiment, but increase cost, system complexity, and hinder scalability due to hardware dependencies.

Vision-based teleoperation offers a scalable, low-cost alternative, enabling natural, expressive control via cameras alone. By exploiting human hand dexterity, it supports rich interaction without additional hardware.

Generalized teleoperation typically comprises two key components: hand pose retargeting and wrist pose retargeting. Hand pose retargeting [13] maps the articulation of a human hand to a robotic counterpart by detecting joint positions and generating corresponding commands. Different robotic hand configurations often require distinct, task-specific kinematic mappings. Wrist pose retargeting is more general by controlling the end-effector's pose via 6-DoF human wrist pose [11]. However, estimating wrist depth from monocular RGB images remains a fundamental challenge.

Various methods have been developed to enhance wrist pose estimation. Multi-camera systems [13] improve accuracy via triangulation but require precise calibration and are sensitive to camera movement. Glove-based systems [16] directly measure joint angles, reducing ambiguity at the cost of increased complexity. RGB-D sensors [17] provide depth information but degrade under occlusion and are less accessible than standard RGB cameras. Recently, learning-based methods [11] using neural networks have shown promise in monocular depth estimation but still require large datasets or extensive training.

These limitations motivate a central research question: how can we achieve accurate and generalizable teleoperation using only monocular RGB input, without increasing system complexity?

To address this, we present **OAT** (**O**ptimization-based h**A**nd pose retargeting and wris**T** pose estimation, illustrated in Fig. 1), a lightweight and hardware-agnostic teleoperation system. OAT employs an optimization-based approach for

[1] School of Computer Science, University of Leeds, Leeds, LS2 9JT, UK {sclch, K.Yao, scsmrd}@leeds.ac.uk
[2] School of Artificial Intelligence, Shanghai Jiao Tong University, China {lipeng.chen@sjtu.com}
M. Dogar was supported by the UK Engineering and Physical Sciences Research Council [EP/V052659/1]
[†] Corresponding author

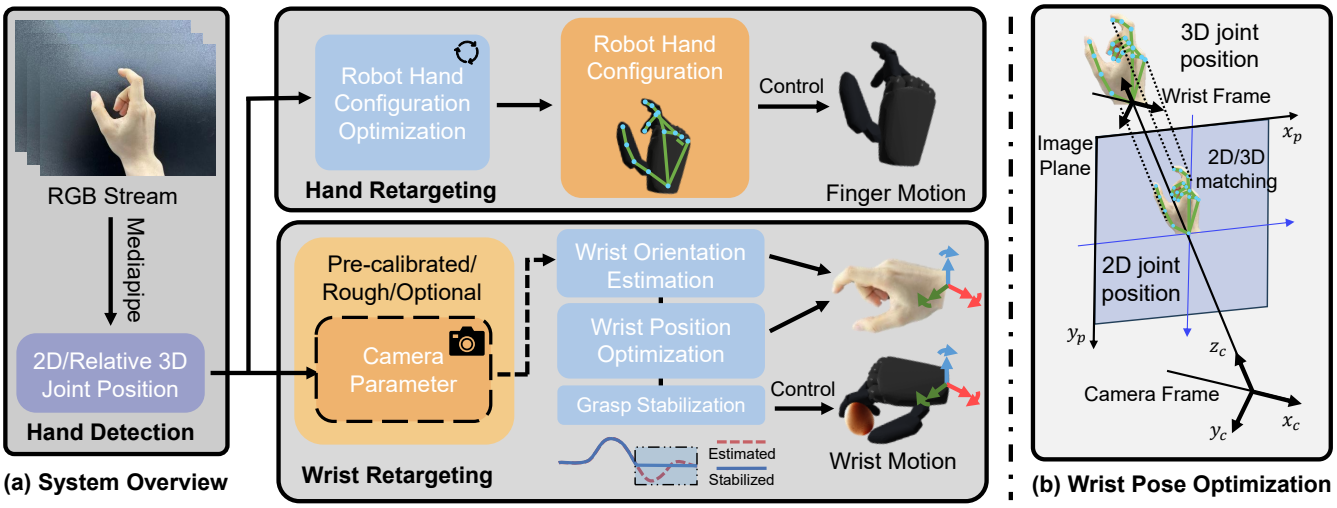

Fig. 2: (a) System Overview. The proposed teleoperation system requires only an RGB camera. The video stream is first processed by Mediapipe [15] to extract 2D and relative 3D hand joint positions. Subsequently, the control pipeline splits into two streams: Hand Retargeting, which maps the human hand pose to the robot hand pose, and Wrist Retargeting, which estimates the wrist's position and orientation, enabling the robot's joints to achieve the corresponding end-effector pose. (b) Wrist pose optimization mechanism. By matching the relative 3D hand joint position in the camera frame and the 2D hand joint position in the image plane, the wrist pose can be inferred.

wrist pose estimation, enabling precise control using only a single RGB camera. Integrated with finger pose retargeting, it supports a wide range of robotic platforms, including both single-arm and multi-arm configurations. The system is easy to deploy, requiring no specialized hardware such as depth sensors, gloves, or multi-camera setups. OAT achieves high wrist pose estimation accuracy and operates faster than existing vision-based teleoperation methods.

With OAT, one of our goals is to enable robot teleoperation with minimal setup, e.g., using a simple laptop camera or a webcam. However, such minimal setups can often come with uncalibrated/inaccurate camera intrinsic parameters. In our experiments, we evaluate the performance of OAT under such conditions and show that it performs robustly even with inaccurate camera intrinsics, but that its performance further improves with camera calibration. Additionally, our framework enables camera self-calibration by jointly optimizing intrinsics from a single hand-containing RGB image.

Finally, integrated with OAT, we introduce a grasp stabilization mechanism that maintains end-effector consistency during grasping actions. A common problem with teleoperation-based grasping is that, finger movements (i.e., change in hand pose) of the human during grasping can cause the wrist pose estimation to change, even if slightly, which results in unwanted robot wrist movements during precise grasping actions. OAT addresses this issue by detecting intervals of large hand-pose change, and smoothing the wrist pose estimation during these intervals.

Section III describes the system architecture, including the retargeting pipeline, grasp stabilization strategy, and camera calibration method. Section IV presents comprehensive evaluations, including qualitative and quantitative results on wrist pose accuracy, runtime comparisons, grasp stability analysis, camera calibration performance and real-world experiments. We evaluate performance both using the Freihand dataset [18] and on our own robotic platform.

II. RELATED WORK

Accurate estimation of hand and wrist pose is fundamental to general-purpose teleoperation. Simple systems like spacemouse [1] or Roboturk [3] are build-free and easy to use, but they lack sufficient DoF and cannot support fine-grained manipulation.

Existing vision-based systems typically rely on either extra devices, multiple cameras or learning-based modules. For example, Open-Teach [4], Open-TeleVision [6], DexCap [5], and Dexpilot [13], improve usability with extra devices, but require substantial setup and are not always platform-agnostic. To reduce hardware requirements, several works leverage monocular RGB cameras. Telekinesis [14] and AnyTeleop [11] apply FrankMocap [19] or deep networks for hand tracking, enabling control without extra devices. However, these approaches often suffer from lower control frequencies, increased system complexity, and depth estimation errors.

Moreover, while works like TeachNet [12], Mosbach et al. [20], and Shaw et al. [21] advance hand motion imitation, many of them lack support for general teleoperation or lightweight deployment on laptops. Despite progress, few existing methods simultaneously achieve build-free and depth-free teleoperation with general applicability and direct usability.

To address these limitations, we propose OAT, an optimization-based teleoperation framework that enables accurate wrist pose estimation using only a single RGB camera.

Our method is build-free, low-cost, and open-source, making it suitable for both research and large-scale deployment.

III. METHOD

An overview of OAT is presented in Fig. 2. OAT utilizes only an RGB camera, such as one integrated into a laptop. The video stream is first processed by MediaPipe [15] to extract 2D and relative 3D hand joint positions. The control pipeline is then split into two streams: hand retargeting, which maps the human hand pose to the robot's hand, and wrist retargeting, which estimates the wrist's position and orientation, allowing the robot arm's joints to achieve the desired end-effector pose. In this section, we provide detailed descriptions of each component.

A. Hand Pose Retargeting

As illustrated in Fig. 2 (b), we first extract 21 2D hand joint positions v_t^i on the image plane and 3D hand joint positions v_t^w in the wrist frame using MediaPipe. The wrist frame shares the same orientation as the camera frame while its translation is different. For hand pose retargeting, only the relative 3D hand joint positions v_t^w are required.

The process of hand pose retargeting has been extensively studied, typically formulated as an optimization problem [13], [22]. The objective is to minimize the difference in key joint positions between the human hand and the robot hand. Different end-effectors, such as grippers, three-finger hands, or five-finger hands, require specific joint index mappings. Following the setting of AnyTeleop [11], we construct the optimization problem as follows:

$$\arg\min_{q_t} \sum_{i=0}^{N} \left\| \alpha_1 {}^i v_t^w - f^i(q_t) \right\|^2 + \beta_1 \|q_t - q_{t-1}\|^2, \quad \text{s.t. } q_l \le q_t \le q_u, \tag{1}$$

where the first term retargets the human hand pose to the robot hand pose, $f^i(q_t) \in \mathbb{R}^3$ computes the 3D position of the i^{th} robot hand joint relative to the robot wrist at timestep t using forward kinematics, and ${}^i v_t^w$ denotes the 3D position of the corresponding human hand joint position relative to the wrist at timestep t. $q_t \in \mathbb{R}^N$ is a vector of the N actuated joint angles of the robot hand at timestep t, α_1 is a scaling factor for robot hand size. The second term ensures smooth robot hand movement by penalizing changes in joint angles. β_1 is a weight term balancing accuracy and smoothness. The bounds q_l and q_u represent the lower and upper joint limits, imposing hard constraints during optimization.

B. Wrist Pose Estimation from RGB Image

The wrist pose estimation consists of position and rotation estimation. The wrist rotation is inferred from the 3D relative positions of the hand joints, while the wrist position requires matching 2D and 3D hand joint positions.

To estimate wrist rotation in the camera frame, a representative palm plane is computed by fitting a geometric plane to five anatomical landmarks: the wrist and the metacarpophalangeal (MCP) joints of the index, middle, ring, and pinky fingers. The plane's normal vector and the in-plane direction from the wrist to the middle finger MCP joint define the hand rotation, with subsequent axes derived orthogonally.

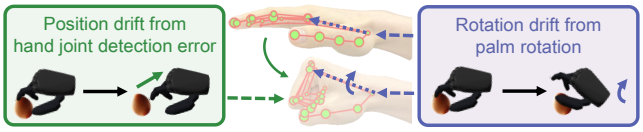

Fig. 3: In natural grasping [23], human hand would induce a rotation in the wrist, in teleoperation this would cause the robot end-effector to rotate unwillingly. On the other hand, different hand poses have different hand joint detection error caused by occlusion, which may bring pose drift in teleoperation.

The wrist position estimation is independent of rotation estimation. Given the camera's intrinsic parameters (estimated or calibrated), we optimize the wrist position by minimizing the projection error between the relative 3D hand joint positions and the 2D hand joint positions in the image plane:

$$\arg\min_{p_t^c} \sum \left\| v_t^i - \theta_c(v_t^w + p_t^c) \right\| + \beta_2 \left\| p_t^c - p_{t-1}^c \right\|^2 \tag{2}$$

where θ_c is the camera intrinsic matrix, v_t^w denotes the joints position in the wrist frame and p_t^c denotes wrist position in the camera frame at time t, respectively. $\theta_c(v_t^w + p_t^c)$ maps the absolute 3D hand joint positions in the camera frame to the image plane. To ensure temporal continuity, a wrist velocity penalty weighted by β_2 is included. The wrist position estimation requires a coarse initialization to avoid division by zero when the depth is zero. We set it to $0.5m$ in all experiments for consistency, which would be updated after the first round of optimization; the actual wrist position in the camera frame is estimated by the system, and teleoperation begins once it stabilizes.

C. Grasp Stabilization

Ideally, teleoperation can be achieved with hand retargeting and wrist pose estimation, enabling data collection. However, during implementation, we observed that when the operator attempts to grasp an object, the robot end-effector shows position and rotation drift. The position drift is caused by estimation errors under occlusion, which can be mitigated by using a more robust hand detection module. During grasping, the hand naturally bends at the palm due to its arched structure [23], causing coordinated movement between the palm and fingers, as shown in Fig. 3.

While signal smoothing technique (low-pass filtering) has been applied to our estimated pose to reduce jitter, we find that they alone are insufficient to suppress large and unintended fluctuations in the robot end-effector, especially during grasping tasks. To address drift in wrist pose estimation, we introduce a bias term p_{bias}, which is added to the robot wrist control as compensation. This bias term compensates for drifts by considering both position and rotation, where rotation is represented using Euler angles. Specifically, when the hand pose changes at a speed exceeding a predefined

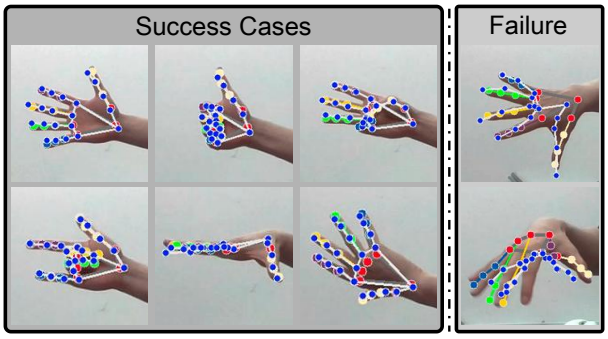

Fig. 4: Qualitative wrist pose optimization results (images are cropped to show details of projection). Colored dots (excluding blue) represent 2D hand pose estimations from MediaPipe in the image plane, while blue dots indicate joint positions projected using the estimated wrist pose. The coincidence of points from two frames shows wrist pose optimization performance.

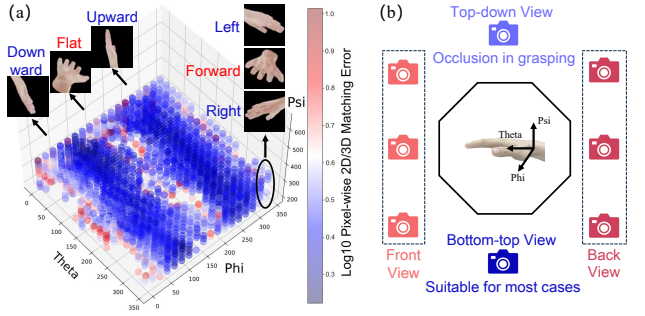

Fig. 5: (a) We present 2D/3D hand joint matching errors from different hand views by images rendered in Blender [25]. Blue dot means small error, red dot means large error; in point-sparse areas MediaPipe fails to detect. (b) The best perspectives are the top-down view and the bottom-up view.

threshold, we record the wrist pose prior to the change and update the bias term accordingly [24]:

$$p_{\text{bias}} \leftarrow p_{\text{bias}} + \mathbf{1}\big(\text{count}_{\psi(q_t - q_{t-1})}\big)(p_t - p_{t-1}) \tag{3}$$

where p_{bias} represents the accumulated wrist pose bias, $\psi(q_t - q_{t-1})$ indicates whether a significant change in hand pose is occurring, and $\text{count}_{\psi(q_t - q_{t-1})}$ tracks the duration of static states. Due to inherent noise and error in the hand detection module, maintaining a counter to reliably identify the actual state is critical. The accumulation of bias is necessary because the hand may perform tasks within certain poses or change poses in various ways. The indicator function $\mathbf{1}\big(\text{count}_{\psi(q_t - q_{t-1})}\big)$ determines whether the bias term should be updated and is defined as:

$$\mathbf{1}_A\big(\text{count}_{\psi(q_t - q_{t-1})}\big) = \begin{cases} 1, & \text{if } \text{count}_{\psi(q_t - q_{t-1})} > 10, \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

This formulation ensures that the wrist pose bias is only updated after the hand has maintained a static state for a sufficient duration, thus minimizing the effect of transient noise or minor hand pose adjustments.

D. Camera Calibration

In Section III-B, camera intrinsics are used to project 3D hand joints onto the image plane for wrist position optimization. These parameters, whether precisely calibrated or coarsely estimated, only affect the scale of the retargeted wrist position and have limited influence on teleoperation.

If more accurate wrist position is needed, by jointly optimizing wrist position and camera intrinsics in Eq. 2, we can estimate camera parameters using a single camera. While this estimation can be rough due to uncertainties in hand detection accuracy and variations in hand sizes, it can still be useful for quick system validation or when extracting camera parameters from online images or videos containing human hands. In our experiments, we compared the camera intrinsics estimated with this method with the true calibration values that come with the Freihand dataset.

IV. EVALUATION

To demonstrate the overall performance of our system, we begin by visualizing the results of the wrist pose optimization and assessing the impact of various camera positions on the accuracy of hand joint estimation. Subsequently, we compare the time delay of OAT with that of other vision-based teleoperation systems, where OAT stands out as the fastest, achieving a 30 Hz control frequency even with a standard laptop. We then evaluate OAT's accuracy and its calibration capability using the Freihand Dataset. Finally, we evaluate the OAT system in our lab environment and report its efficiency relative to keyboard and SpaceMouse, which are standard control interfaces in teleoperation tasks.

A. Teleoperation with OAT

Camera Position During evaluation, we found it important to put the camera at a suitable position in our system. The two optimization-based methods (Eq. 1 and 2) are independent of camera position. However, MediaPipe's [15] performance heavily depends on how the human hand appears in the image, i.e., the camera position. We present the qualitative result of the wrist pose estimation in Fig. 4. In the figure, colorful dots (except blue) represent 2D hand pose estimations from MediaPipe in the image plane, while blue dots indicate joint positions projected using the estimated wrist pose using our method.

In success cases, in which MediaPipe generates good 2D and 3D hand joint estimation, the hand joints in the camera frame overlap the hand joints in the image plane. In failure cases, the 3D hand joints present incorrect structure as MediaPipe fails to estimate both 2D and 3D hand joints well. Wrist pose estimation puts two sets of hand joints closely, but not with a good match, which causes errors in wrist pose estimation and results in teleoperation errors.

From experience and as shown in Anyteleop [11], it is better to position the camera on the ground with the hand teleoperating above the camera. To quantitatively validate

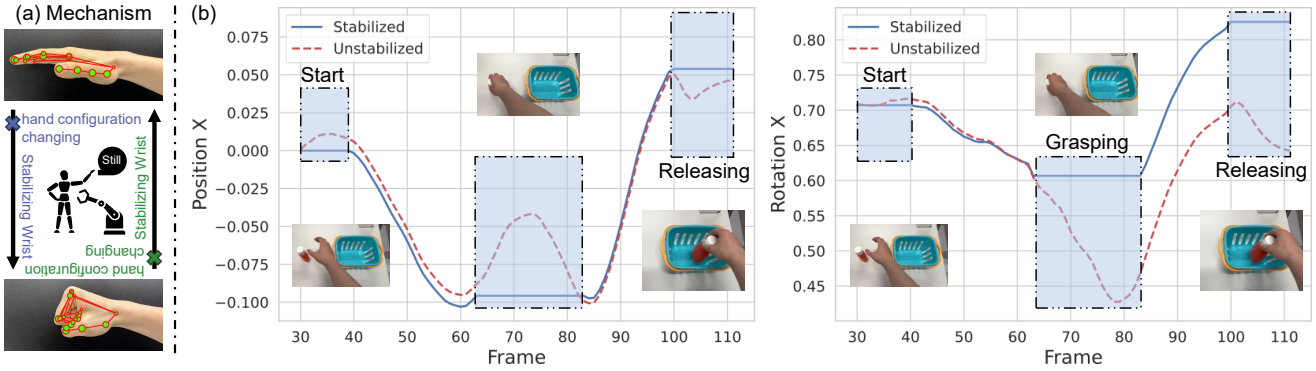

Fig. 6: (a) Grasping stabilization mechanism. (b) We record the estimated wrist pose during a pick-and-place task. Without stabilization, the robot's wrist rotates and lifts due to hand motion and detection errors, leading to grasping and releasing failures. In contrast, with stabilization enabled, the wrist remains stationary during hand pose changes, enabling the robot to successfully complete the task. The stabilized wrist pose ensures smooth, precise, and reliable teleoperation.

which camera positioning performed better, we measured the 2D/3D matching errors from different views. We first built and rendered a realistic human hand in Blender [25] from different views 1.0 m away, presenting the hand images to OAT. The hand was rendered in a common open pose. We use pixel-wise 2D/3D matching error to present the performance, as given below:

$$\text{Error} = \frac{\sum_{i=1}^{N} \| {}^{i}v_t^i - \theta_c({}^{i}v_t^w + p_t^c) \|}{\sum_{i=1}^{N} \| {}^{i}v_t^i \|} \tag{5}$$

Fig. 5 visualizes the matching errors with blue dots representing small errors and red dots indicating larger errors. In regions where no data is displayed, MediaPipe fails to detect the hand joints. Notably, the most accurate perspectives are the top-down view and the bottom-up view, as these angles provide better alignment and more precise detection of hand joints. As top-down view would result in occlusion of finger in manipulation, we suggest putting and fixing the camera in a lower plane.

Grasping Stabilization We present the wrist pose estimation results in a real-world pick-and-place procedure in Fig. 6. In the grasping and releasing phases illustrated, the operator needs to keep the hand stationary to grasp the object or release the object to a desired position. Without stabilization, the robot lift and rotate when the hand changed pose (grasping/releasing), leading to task failures and potential collisions with the environment. With stabilization, the hand remains static during both grasping and releasing phases, ensuring smooth state transitions and reducing the risk of operational errors.

B. Time Delay Measurement

In robot teleoperation, time delay is a critical factor affecting feedback and efficiency. We evaluated OAT's efficiency on two laptops and two desktops, with results summarized in Table I. The two laptops (A and B), achieved approximately 30 Hz control frequency. The desktops with a better GPU reached about 50 Hz by reducing the time cost of mediapipe. While hand and wrist pose retargeting are relatively quick,

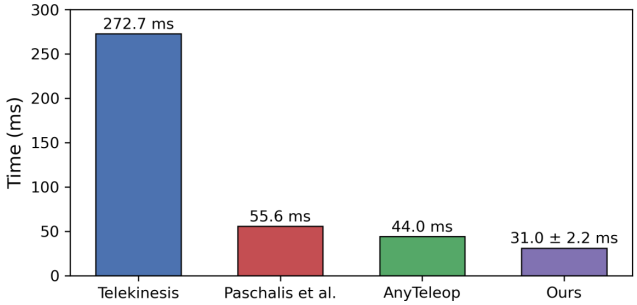

Fig. 7: Runtime comparison of our method with recent state-of-the-art systems. By incorporating our wrist pose optimization technique, our method achieves the fastest runtime. We report the runtime of our method on the worst hardware specification (please see Table. I for faster results on better hardware). Runtime estimates for the other three methods are taken from their respective papers.

TABLE I: OAT Runtime on Different Hardwares

Hardware	Type	Laptop A	Laptop B	Desktop A	Desktop B
	CPU	i9-10980HK	i7-10875H	i7-13700	w7-3465x
	GPU	RTX 3080	GTX 1650	RTX 4060	RTX A6000
Profiling	**Modules**	**Time (ms)**			
	MediaPipe	19	18	16	10
	Hand Retargeting	2.3	2.4	1.5	1.5
	Wrist Retargeting	13	13	6.1	6.1
	Loop Time	31	31	22	17

MediaPipe, which detects hand features, takes the longest time and presents limited optimization potential. The actual delay depends on the device used and the speed of hand motion—faster movements can degrade MediaPipe's accuracy, increasing the optimization time required.

We compared OAT's time delay with several recent vision-based teleoperation systems, as shown in Fig. 7. Telekinesis [14] employs single RGB images like OAT but uses

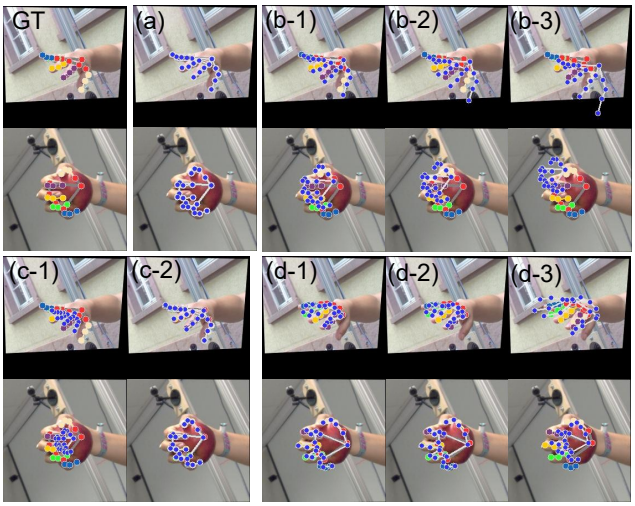

Fig. 8: Qualitative results on the FreiHand dataset under various conditions. GT: Ground truth 2D joint positions. (a) Precise calibration. (b) With calibration errors: 5% (b-1), 10% (b-2), 20% (b-3). (c) Without (c-1) and with (c-2) calibration optimization. (d) MediaPipe results: with precise calibration (d-1), without (d-2), and with (d-3) calibration optimization.

FrankMocap [19] for body and hand pose estimation, which is time-consuming. Panteleris et al. [26], although not a teleoperation system, also use single RGB images for hand pose estimation. Their method optimizes wrist pose through hand kinematics, resulting in lower processing speed. AnyTeleop, which utilizes depth cameras and neural networks for wrist depth estimation, is relatively slower than OAT due to its depth-based processing approach. Runtime estimates for the other three methods are obtained from their respective papers, as evaluation on the same computer is not feasible due to their closed-source or compilation error. The configurations reported in their works are as follows: Telekinesis employed an AMD 3960X CPU and dual NVIDIA 3080 Ti GPUs; Panteleris et al. used a workstation with an Intel i7 CPU and an NVIDIA GTX 1070 GPU; and AnyTeleop's results were based on a desktop with an NVIDIA RTX 3090 GPU and an Intel i9-10980XE CPU.

C. Accuracy and Calibration in Freihand Dataset

To evaluate the precision of the proposed wrist pose estimation method, we conducted experiments on the Freihand dataset [18]. This dataset includes RGB images of human hands, 2D and 3D hand pose labels, and camera intrinsic parameters. The discrete nature of the RGB images in the dataset poses a challenge to our method, as continuous image streams can improve the stability and accuracy of optimization results.

We performed nine tests on the Freihand dataset, summarized in Table. II, and the qualitative result is shown in Fig. 8. The evaluation focused on wrist position estimation precision and comparing it with the ground truth. We utilized the provided 2D and 3D hand pose labels and camera intrinsic parameters from the dataset. The tests were conducted in four distinct settings:

TABLE II: Freihand evaluation result

Metrics	Unrecognized Percentage	Wrist Position Error (m)	Calibration Error
Precise Calibration	-	9.502e-7	0.00%
5% Calibration Error	-	3.120e-2	5.47%
10% Calibration Error	-	6.281e-2	10.97%
20% Calibration Error	-	1.291e-1	20.22%
without Calibration Optimization	-	2.685e-1	62.31%
with Calibration Optimization	-	1.336e-1	14.81%
Mediapipe with Precise Calibration	84.67%	1.141e-1	0.00%
Mediapipe without Calibration Optimization	84.67%	7.647e-2	8.194%
Mediapipe with Calibration Optimization	84.67%	4.277e-2	31.77%

(a) Precise calibration: We directly used the ground truth camera intrinsic parameters and hand joint position.
(b) 5-20% calibration error: We introduced noise into the camera intrinsic parameters, with the error percentage calculated relative to the ground truth values.
(c) Wrist position estimation with/without Camera Optimization: Without ground truth camera parameter, we assumed no prior knowledge of camera parameters, setting the focal length to 300mm and the optical center to the image center. For the setting with camera optimization, we performed camera calibration to estimate camera intrinsic parameters while simultaneously estimating wrist position.
(d) Wrist position estimation with Mediapipe: Instead of using ground truth hand pose data, we use Mediapipe to detect hand in the image. Other settings are the same as (c).

From the results of (a) and (b), we observed a nearly linear relationship between increasing calibration errors and wrist position estimation errors. With precise calibration, the wrist position estimation was highly accurate. As the calibration error increased, the optical center was around the image center, and the focal length error deviated proportionally to the error percentage.

For the setting (c), the camera intrinsic parameter error reached 62.31%, resulting in a 26.85 cm wrist position error. With gound truth joint data and calibration optimization, the calibration error was reduced to approximately 15%, decreasing the wrist position estimation error to 13.36 cm—comparable to the result under a 20% calibration error.

In setting (d), we evaluated our wrist pose estimation method using MediaPipe. Instead of ground truth 2D and 3D hand pose data, we used MediaPipe's hand detection results. Both left- and right-hand detectors were employed, but only 15.33% of the images were successfully detected.

Using MediaPipe, we tested three configurations: with precise calibration, without calibration optimization, and with

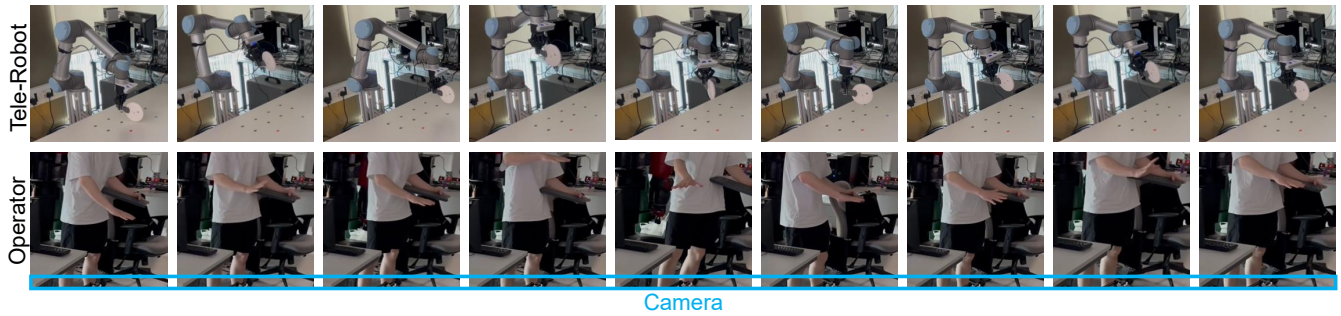

Fig. 9: Qualitative demonstration of teleoperating the UR5 robot with OAT. The robot grasps a plate to better illustrate end-effector rotation. The operator's right hand, captured by a ground-level camera, is retargeted to control the robot, while the left hand operates a keyboard to trigger the start and termination of teleoperation.

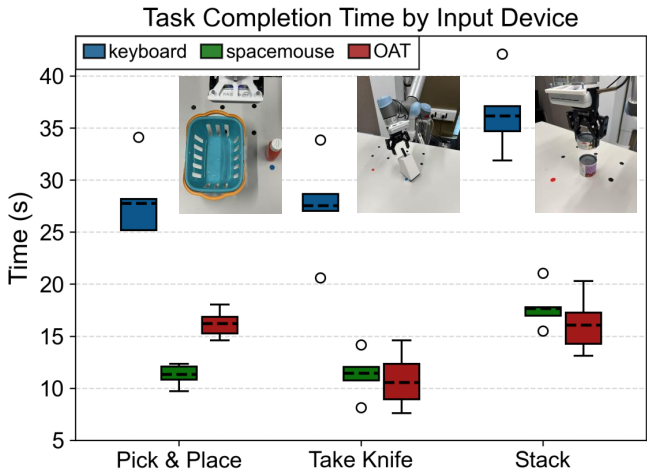

Fig. 10: Task completion time across three real-world tasks. Both OAT and the spacemouse substantially outperform the keyboard, while OAT achieves comparable performance to the spacemouse in rotation- and precision-critical tasks (*take knife* and *stack*).

calibration optimization. Contrary to expectations, precise calibration produced the largest wrist position estimation error (11.41 cm). Calibration optimization reduced the wrist position error from 7.647 cm to 4.277 cm, falling within an acceptable range. Notably, the calibration error in the MediaPipe without calibration optimization setting was only 8.19%, compared to 62.31% when not using MediaPipe, indicating that MediaPipe's detection works well with similar camera settings. After calibration optimization, the calibration error increased, likely due to the hand pose estimation error of MediaPipe.

D. Real-world Testing

Before presenting the quantitative evaluation, we first provide a qualitative example to illustrate the basic functionality of OAT in a real-world setting (Fig. 9). The example shows the system's ability to map human hand motions to precise robot control.

We conducted three quantitative experiments to evaluate the task efficiency of OAT (Fig. 10, other views of experiments in Fig. 1). The evaluation covered three representative tasks: *pick and place*, targeting fundamental spatial positioning; *take knife*, emphasizing orientation-sensitive grasping; and *stack*, requiring precise alignment and stable placement. Each task was executed ten times using three control modalities: keyboard, spacemouse, and OAT.

In the *pick and place* task, both OAT and the spacemouse achieved substantially faster completion times than the keyboard, with the spacemouse maintaining a slight advantage over OAT. In contrast, for the *take knife* and *stack* tasks—which demand rotational alignment and high precision—OAT performed comparably to the spacemouse and consistently outperformed the keyboard.

V. CONCLUSION

In this paper, we propose a lightweight retargeting teleoperation system, OAT. Our system leverages open-source MediaPipe combined with an optimization-based approach to achieve single-RGB teleoperation, and its efficiency is further validated through real-world experiments. By formulating the wrist pose estimation problem as an optimization task, our system achieves both efficiency and precision for manipulation. For applications requiring precise teleoperation and absolute wrist pose accuracy, we introduce grasp stabilization and camera calibration as supporting tools.

The system can operate without precise camera calibration and provides rough calibration during teleoperation. While depth–focal length ambiguity can affect focal length estimation when using hand-based cues, our optimization reduces intrinsic errors and ensures that relative precision, which matters more than absolute metric accuracy, remains acceptable in practice. The OAT formulation is general and not limited to the specific robotic platform used here; it can be easily adapted to arbitrary end-effectors by modifying the joint mapping.

Although our implementation accuracy relies on MediaPipe for 2D/3D hand joint detection, this is not an inherent limitation of OAT. Our optimization based wrist pose estimation approach can be easily integrated with any hand pose estimator. Improved upstream detectors would directly lead to higher accuracy and robustness.

REFERENCES

[1] J. Nádvorník and P. Smutný, "Remote control robot using android mobile device," in *Proceedings of the 2014 15th International Carpathian Control Conference (ICCC)*, 2014, pp. 373–378.

[2] M. Zare, P. M. Kebria, A. Khosravi, and S. Nahavandi, "A survey of imitation learning: Algorithms, recent developments, and challenges," 2024.

[3] A. Mandlekar, Y. Zhu, A. Garg, J. Booher, M. Spero, A. Tung, J. Gao, J. Emmons, A. Gupta, E. Orbay *et al.*, "Roboturk: A crowdsourcing platform for robotic skill learning through imitation," in *Conference on Robot Learning*. PMLR, 2018, pp. 879–893.

[4] A. Iyer, Z. Peng, Y. Dai, I. Guzey, S. Haldar, S. Chintala, and L. Pinto, "Open teach: A versatile teleoperation system for robotic manipulation," PMLR, pp. 2372–2395, 2025.

[5] C. Wang, H. Shi, W. Wang, R. Zhang, L. Fei-Fei, and K. Liu, "Dexcap: Scalable and portable mocap data collection system for dexterous manipulation."

[6] X. Cheng, J. Li, S. Yang, G. Yang, and X. Wang, "Open-television: Teleoperation with immersive active visual feedback," in *8th Annual Conference on Robot Learning*, 2024.

[7] K. Darvish, L. Penco, J. Ramos, R. Cisneros, J. Pratt, E. Yoshida, S. Ivaldi, and D. Pucci, "Teleoperation of humanoid robots: A survey," pp. 1706–1727, 2023.

[8] L. Meng, J. Liu, W. Chai, J. Wang, and M. Q.-H. Meng, "Virtual reality based robot teleoperation via human-scene interaction," *Procedia Computer Science*, vol. 226, pp. 141–148, 2023, proceedings of International Conference on Biomimetic Intelligence and Robotics. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S187705092301815X

[9] L. Chen, Z. J. Hu, Y. Huang, E. Burdet, and F. R. y Baena, "Human robot shared control in surgery: A performance assessment," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 15 165–15 171.

[10] S. Dass, W. Ai, Y. Jiang, S. Singh, J. Hu, R. Zhang, P. Stone, B. Abbatematteo, and R. Martín-Martín, "Telemoma: A modular and versatile teleoperation system for mobile manipulation," in *2nd Workshop on Mobile Manipulation and Embodied Intelligence at ICRA 2024*, 2024.

[11] Y. Qin, W. Yang, B. Huang, K. Van Wyk, H. Su, X. Wang, Y.-W. Chao, and D. Fox, "Anyteleop: A general vision-based dexterous robot arm-hand teleoperation system," 2023.

[12] S. Li, X. Ma, H. Liang, M. Görner, P. Ruppel, B. Fang, F. Sun, and J. Zhang, "Vision-based teleoperation of shadow dexterous hand using end-to-end deep neural network," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 416–422.

[13] A. Handa, K. Van Wyk, W. Yang, J. Liang, Y.-W. Chao, Q. Wan, S. Birchfield, N. Ratliff, and D. Fox, "Dexpilot: Vision-based teleoperation of dexterous robotic hand-arm system," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 9164–9170.

[14] A. Sivakumar, K. Shaw, and D. Pathak, "Robotic telekinesis: Learning a robotic hand imitator by watching humans on youtube," in *Workshop on Learning from Diverse, Offline Data*, 2022.

[15] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, J. Lee, W.-T. Chang, W. Hua, M. Georg, and M. Grundmann, "Mediapipe: A framework for building perception pipelines," 2019. [Online]. Available: https://arxiv.org/abs/1906.08172

[16] H. Liu, X. Xie, M. Millar, M. Edmonds, F. Gao, Y. Zhu, V. J. Santos, B. Rothrock, and S.-C. Zhu, "A glove-based system for studying hand-object manipulation via joint pose and force sensing," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 6617–6624.

[17] Q. Vuong, Y. Qin, R. Guo, X. Wang, H. Su, and H. Christensen, "Single rgb-d camera teleoperation for general robotic manipulation," 2021. [Online]. Available: https://arxiv.org/abs/2106.14396

[18] C. Zimmermann, D. Ceylan, J. Yang, B. Russel, M. Argus, and T. Brox, "Freihand: A dataset for markerless capture of hand pose and shape from single rgb images," in *IEEE International Conference on Computer Vision (ICCV)*, 2019. [Online]. Available: "https://lmb.informatik.uni-freiburg.de/projects/freihand/"

[19] Y. Rong, T. Shiratori, and H. Joo, "Frankmocap: Fast monocular 3d hand and body motion capture by regression and integration," 2020. [Online]. Available: https://arxiv.org/abs/2008.08324

[20] M. Mosbach, K. Moraw, and S. Behnke, "Accelerating interactive human-like manipulation learning with gpu-based simulation and high-quality demonstrations," in *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*. IEEE, 2022, pp. 435–441.

[21] K. Shaw, Y. Li, J. Yang, M. K. Srirama, R. Liu, H. Xiong, R. Mendonca, and D. Pathak, "Bimanual dexterity for complex tasks," in *8th Annual Conference on Robot Learning*, 2024.

[22] Y. Qin, Y.-H. Wu, S. Liu, H. Jiang, R. Yang, Y. Fu, and X. Wang, "Dexmv: Imitation learning for dexterous manipulation from human videos," Springer, pp. 570–587, 2022.

[23] N. J. Jarque-Bou, A. Scano, M. Atzori, and H. Müller, "Kinematic synergies of hand grasps: a comprehensive study on a large publicly available dataset," *Journal of neuroengineering and rehabilitation*, vol. 16, pp. 1–14, 2019.

[24] C. Movement, "A minimal intervention principle for," in *Advances in Neural Information Processing Systems 15: Proceedings of the 2002 Conference*, vol. 15. MIT Press, 2003, p. 27.

[25] B. O. Community, *Blender - a 3D modelling and rendering package*, Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. [Online]. Available: http://www.blender.org

[26] P. Panteleris, I. Oikonomidis, and A. Argyros, "Using a single rgb frame for real time 3d hand pose estimation in the wild," IEEE, pp. 436–445, 2018.