



Stargazer: An Interactive Camera Robot for Capturing How-To Videos Based on Subtle Instructor Cues

Jiannan Li
University of Toronto
Toronto, Canada

Bryan Wang
University of Toronto
Toronto, Canada

Angela Yang
University of Toronto
Toronto, Canada

Mauricio Sousa
University of Toronto
Toronto, Canada

Paula Akemi Aoyau
University of Toronto
Toronto, Canada

Ravin Balakrishnan
University of Toronto
Toronto, Canada

Tovi Grossman
University of Toronto
Toronto, Canada

Karthik Mahadevan
University of Toronto
Toronto, Canada

Nicole Yu
University of Toronto
Toronto, Canada

Anthony Tang
Singapore Management University
Singapore

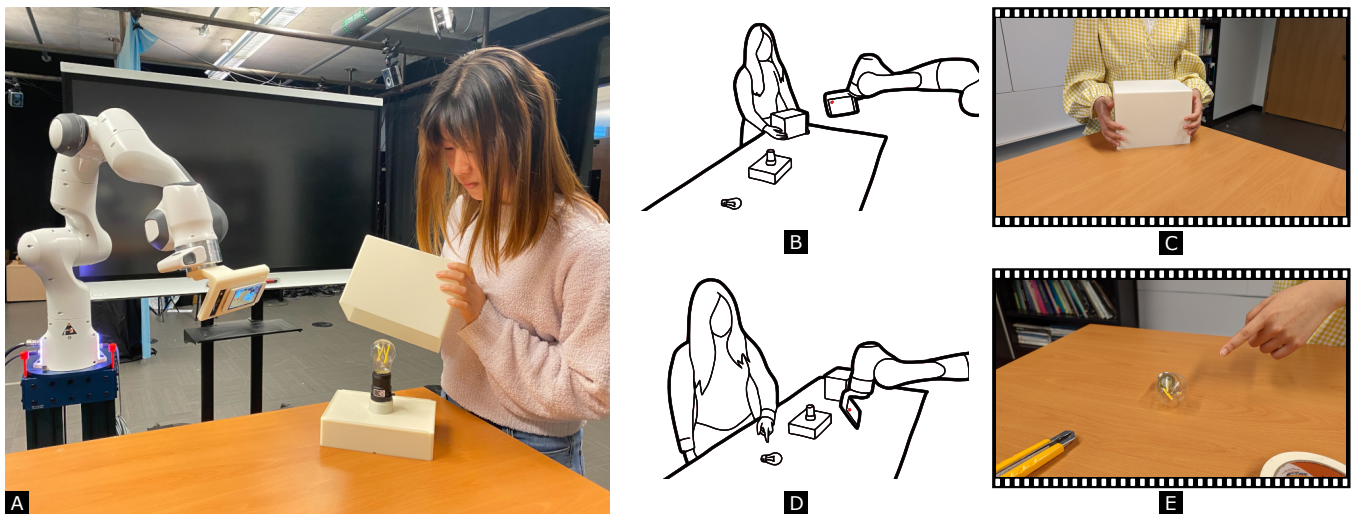


Figure 1: (A) Stargazer is a novel approach for how-to video creation using a camera robot. (B & C) Stargazer's camera automatically follows the region of interesting, such as the hands of instructor who is holding the cover for a lamp. (D & E) Stargazer also detects the instructor's signals, such as pointing gestures, and responds by focusing on the light bulb.

ABSTRACT

Live and pre-recorded video tutorials are an effective means for teaching physical skills such as cooking or prototyping electronics. A dedicated cameraperson following an instructor's activities can improve production quality. However, instructors who do not

have access to a cameraperson's help often have to work within the constraints of static cameras. We present Stargazer, a novel approach for assisting with tutorial content creation with a camera robot that autonomously tracks regions of interest based on instructor actions to capture dynamic shots. Instructors can adjust the camera behaviors of Stargazer with subtle cues, including gestures and speech, allowing them to fluidly integrate camera control commands into instructional activities. Our user study with six instructors, each teaching a distinct skill, showed that participants could create dynamic tutorial videos with a diverse range of subjects, camera framing, and camera angle combinations using Stargazer.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CHI '23, April 23–28, 2023, Hamburg, Germany

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-9421-5/23/04...\$15.00

<https://doi.org/10.1145/3544548.3580896>

CCS CONCEPTS

• **Human-centered computing** → **Interactive systems and tools**; • **Computer systems organization** → **Robotics**.

KEYWORDS

cameras, robots, instructional videos

ACM Reference Format:

Jiannan Li, Mauricio Sousa, Karthik Mahadevan, Bryan Wang, Paula Akemi Aoyai, Nicole Yu, Angela Yang, Ravin Balakrishnan, Anthony Tang, and Tov Grossman. 2023. Stargazer: An Interactive Camera Robot for Capturing How-To Videos Based on Subtle Instructor Cues. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23)*, April 23–28, 2023, Hamburg, Germany. ACM, New York, NY, USA, 16 pages. <https://doi.org/10.1145/3544548.3580896>

1 INTRODUCTION

People use how-to videos, both live and pre-recorded, to learn new physical skills [37], ranging from repairing a broken keyboard to learning digital fabrication [42]. In most how-to videos for physical skills, the instructor demonstrates step-by-step how to complete the task [14]. As these steps may involve activities at varying locations in varying levels of detail, a single, fixed camera often cannot record every step with the desired clarity [39]. This necessitates frequent changes to camera parameters, including viewpoints, angles, and zoom levels.

Professional video productions, such as cooking and home improvement shows, employ several dedicated camera operators who actively re-position cameras and adjust their parameters in response to the instructor's actions. However, such resources are not available to most instructors; instead, these rely on one or more pre-configured fixed cameras. Although fixed camera setups can be re-configured during recording, instructors need to stop what they are demonstrating (e.g., chopping vegetables) to manipulate the camera. This disrupts the demonstration, increasing the instructor's workload. It also requires more post-processing to combine clips filmed with different camera setups.

Both filmmakers and researchers have explored the idea of camera-manipulating robots as an alternative to human operators [1, 27]. Recent camera robots (predominantly drones) can autonomously track moving subjects [6, 24, 41]. However, it remains a challenge for the user being filmed to control the camera robots' behaviors while performing other activities, such as demonstrating a physical process. Conventional interfaces for robot control employ joysticks [52], gestures [49], and speech [16]—all of which require dedicated input actions—that disrupt instruction delivery. If not edited out, such disruptions might split audience's attention and hinder learning [9], but post-processing adds to instructors' efforts. Recent user interface research has explored triggering on-screen visual effects through presenters' gestures and speech [22, 32, 48] that are part of the presentations. Our approach in this work is to: (1) identify the kinds of camera shots that how-to videos use and (2) direct camera operations in a non-disruptive manner by relying on the communicative signals that instructors already use to address their *audience* during demonstrations. For instance, an instructor may point to a part of an object to emphasize it, use speech to guide the audience's attention, or wave to introduce themselves.

We present Stargazer, a novel approach that uses a camera robot to capture dynamic how-to videos for tabletop-scale physical tasks (Figure 1A). Instructors can use subtle signals to control the high-level behavior of the camera, such as the current region of interest and camera framing, in real-time. Stargazer's camera autonomously tracks and captures the region of interest (Figure 1B-E). Note that our goal is not to build a fully automated robot that understands every nuance of human communicative behavior; instead, we aim to explore a vocabulary of controls that can support the necessary camera work needed to create informative tutorial videos and can be fluidly blended into normal instruction activities.

Based on an analysis of 50 how-to videos under three popular categories, we identified three common types of shots and three types of camera parameter control that the robot should support. Instructors control shot types and camera parameters through gestures and speech that they commonly use to guide the attention of their audience. Stargazer locates the subject of interest in each type of shot and smoothly tracks them.

We invited six participants, including two professional filmmakers, to capture how-to videos with Stargazer for a physical task of their expertise. All participants were able to create videos without needing any dedicated controls other than the cues afforded by Stargazer and were satisfied with the quality of the video produced. Based on participants' think-aloud comments on the produced videos and post-study interview data, we identify opportunities and challenges for collaborative video capture with robots.

2 RELATED WORK

Our research builds on prior work on how-to videos, camera robots, and human cues for robot control. Our interaction design is inspired by recent progress in performance-driven interactive presentations. Below we review related work.

2.1 How-To Videos

How-to videos are commonplace on online video and livestreaming websites and provide convenient ways to learn physical tasks [14, 17, 37]. Research in human-computer interaction (HCI), cognitive science, and education technologies has studied how attributes of how-to videos, such as perspectives [26], pace [58], and instructor identities [23], could impact learning. User interface research has specifically investigated novel interfaces for exploring [10], navigating [11, 61], annotating [29], and editing [12] how-to videos. Our work builds on these previous works on how-to videos but aims to facilitate filming these videos instead of their post-processing or consumption.

2.2 Interactive Presentations

Researchers have explored how the performance of the presenter, including speech [32], gesture [22, 48], and sketching [43, 55], can drive real-time on-screen graphics to make live or recorded presentations more informative and engaging. A particularly elegant aspect of these interactive presentation interfaces is that they allow presenters to merge graphics effect controls into content delivery. RealityTalk [32] leveraged keywords in verbal presentations to trigger visual augmentations. Saquib et al. [48] enabled presenters to directly manipulate graphical elements and effects with gestures

and changes in postures. Hall et al. [22] also studied the use of gestures, focusing on modifying data visualizations. Our user interactions with Stargazer are inspired by how prior work fluidly fits effect controls into presenter performance by leveraging gestures and speech. Instead of controlling digital artifacts captured by a fixed camera, Stargazer focuses on controlling a dynamic camera with instructor actions.

2.3 Human Cues for Human-Robot Interaction

Prior research has examined coordination mechanisms that human-human teams use for collaborative physical tasks. These tasks demand joint attention—directing a partner’s attention to an object or area of interest. Human-robot interaction researchers have proposed several methods to aid joint attention. A direct approach can be having users annotate the robots’ sensor input (e.g., camera streams) [51, 60], but this usually requires an additional device. Tracking the human’s gaze enables anticipatory robot planning and execution of actions [2, 4, 5, 25, 40], or can be used to prompt robot takeover when it detects user hesitation [47]. A gesture is another mechanism to direct attention by humans and robots [49], including pointing, presenting, and exhibiting objects while collaborating on physical tasks [20]. Humans can even use language to direct a robot’s attention [56], which has been utilized for hands-free robot programming [16] and commanding a robot during pick-and-place tasks [36]. We note that prior research has typically used either explicit communication mechanisms (i.e., pointing, commands, etc.) or more implicit communication mechanisms (i.e., eye gaze) to manipulate the robot. Stargazer’s approach employs both explicit and implicit communication acts. Still, our approach considers these acts part of the performative dialogue between the presenter and the *audience* rather than between the presenter and the robot.

2.4 Camera Robots

Research in remote assistance and telepresence mostly studied camera robots that were manually operated with GUIs by remote users (e.g., [21, 31, 57, 59]). Recent remote assistance camera robots, such as RemoteCoDe [46], can be controlled implicitly by remote helpers’ face orientation. Another line of work has explored camera robots that react to signals from the people being filmed. Autonomous camera drones can track the human subject and adjust their flight plans based on the subject’s positions [41], postures [24], and actions [6], often informed by cinematography principles [27] or learned preferences [6]. While not filming humans, Rakita et al. [44] designed a method for a robot arm to dynamically capture another arm being teleoperated to assist teleoperation. Although responsive to human signals, these robots mainly do not allow the user being filmed to actively initiate significant behavior changes in the robots, such as changing the subject or camera framing. Stargazer aims to fill this gap by providing instructors with more fine-grained real-time camera behavior controls for filming how-to videos.

3 CHARACTERIZING CAMERA USE IN HOW-TO VIDEOS

The design of Stargazer’s behaviors and interactions was informed by studying the content and visual language of existing how-to videos. We studied 50 videos from three categories of physical skill

instruction with the aim of learning: (1) What are the common types of shots in these videos? and (2) How are framing, angles, and movement used in how-to videos by creators?

3.1 Video Corpus

Our video corpus contains 50 instructional videos, with 15 on Computers and Electronics, 18 on Food and Entertainment, and 17 on Hobbies and Crafts. The average video length is 5 minutes 22 seconds. The average number of views was 2,176,470 (min: 20,824; max: 37,067,768) as of July 2022. We constructed our corpus based on physical tasks on WikiHow¹, a well-known database with thousands of user-produced how-to articles on a wide range of daily tasks, based on strategies used in prior work [38].

To identify classes of physical tabletop-scale tasks, we followed “Popular Categories” on the website, and found three categories: “Computer and Electronics”, “Hobbies and Crafts”, and “Food and Entertainment”. We then randomly surveyed each category to gather 50 how-to articles and used the article titles to search for videos online on YouTube² using incognito mode. We then selected videos based on the following criteria: 1) Responds to the how-to question collected from WikiHow; 2) Recorded in the English language; 3) At most 20 minutes long; 4) Accumulated over twenty thousand views; We used number of views as a proxy to ensure reasonable quality for the sampled videos. A recent survey of instructional and informational video viewers found that less than 10% of their respondents preferred videos longer than 20 minutes [30]. Therefore, we focused on videos at most 20 minutes long to allow our team to analyze a sufficiently diverse dataset within our resource constraints.

3.2 Analysis

We coded the sampled videos at a shot level (884 shots in total, average of 17.68 shots each), focusing on what content was being captured in each shot and how they were communicated through distinct visual language (i.e., angles and framing). We created a bottom-up coding scheme with two major categories: *subject in the shot* and *camera setup* (i.e., angles, framing, and movements). *Subject in the shot* covers what is being shown (e.g. the host, objects). Our coding scheme for *Camera setup* was informed by common cinematography practices [7]. This category relates to the visual language, including camera angle (e.g., high angle, over-the-shoulder), camera framing (e.g., medium, close-up), and camera movements (e.g. pan, truck, orbit). Two researchers created and tested the coding scheme by analyzing five videos together and then worked separately on the remaining videos.

3.3 Subjects of Shots

We observed that how-to video shots focus on a small set of subjects—the instructor, the actions of the instructor, and the objects used in demonstrating the task. The instructor’s actions primarily manifest in the form of manipulating tools and other artifacts with their hands. How-to video shots often feature both the instructor’s hands and the objects being manipulated (*hands+objects*, 53.51% of total video length, Figure 2 A and D). The second category of

¹<https://www.wikihow.com/Main-Page>

²<https://www.youtube.com/>

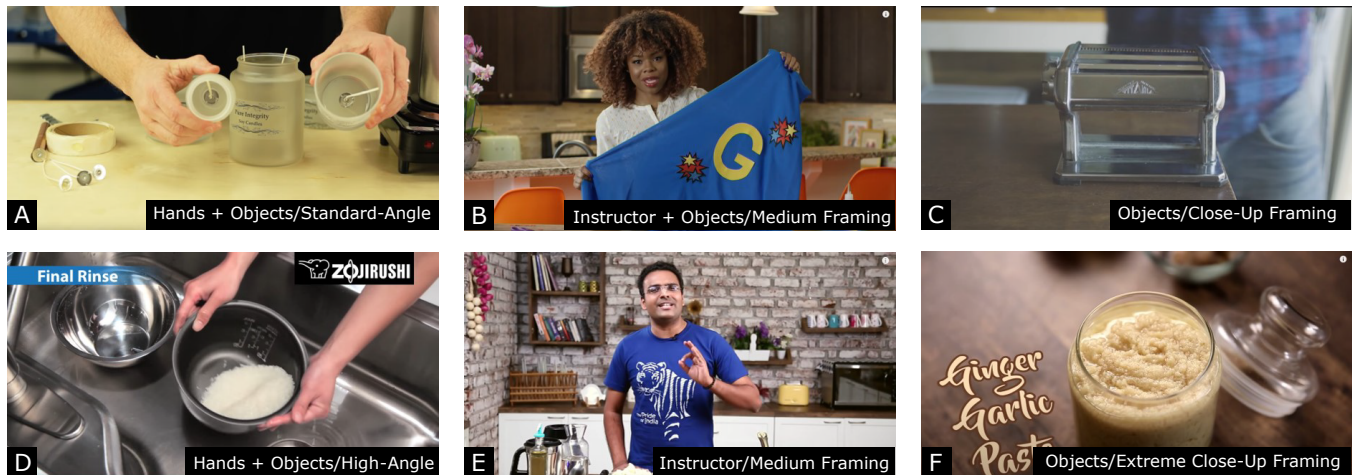


Figure 2: Example frames from the video analysis. (A) a *hands+objects* shot from the standard camera angle (B) an *instructor+objects* shot with medium framing (C) an *objects* shot with close-up framing (D) a *hands+objects* shot with from a high camera angle (E) an *instructor* shot with medium framing (F) an *objects* shot with extreme close-up framing

shots focuses more on the instructor verbally “lecturing” to the audience and thus primarily shows the instructor talking and presenting objects to them (*instructor*, 10.18% of total video length and *instructor+objects*, 11.88% of total video length; see Figure 2 B and E). Finally, 17.65% of the total video length analyzed includes shots featuring objects alone (*objects*; Figure 2 C and F). These shots often show the materials or tools used in the task or the outcome of the process.

3.4 Visual Language

While the visual language in film-making encompasses a myriad of elements, our analysis focused on those relevant to the positioning of cameras, i.e., camera framing, angle, and movements.

3.4.1 Camera Framing. We found that shots showing each type of subject mainly use two types of framing. Shots showing *hands+objects* or *objects* alone predominantly uses close-up framing (79.92% for *hands+objects*, 64.74% for *objects* alone, Figure 2A, C, D), followed by extreme close-up (15.86% for *hands+objects*, 28.21% for *objects* alone, Figure 2 F). Shots focusing on the instructor are filmed more with medium framing (66.67% for the instructor, 87.62% for *instructor+objects*, Figure 2B and E) than with close-up (33.33% for *instructor*, 10.48% for *instructor+objects*).

3.4.2 Camera Angles. When coding camera angles, we used common camera-angle types while adapting them to fit the context of how-to videos. Following cinematography conventions, we labeled shots looking down at the subject (*hands/instructor/objects*) as *high-angle* and those looking up at the subject as *low-angle*. Overhead shots were also included in the *high-angle* category. Shots from the same level as the subject, or slightly above it, were labeled as *standard*. We found that most shots focusing on the instructor were made with a standard angle (98.89% for *instructor* and 82.86% for *instructor+objects*, Figure 2B and E). In contrast, high-angle (22.42%), point-of-view (34.02%), and standard angles (29.57%) are all commonly seen in shots focusing on *hands+objects* and *objects*.

Figure 2A and D show examples of standard-angle and high-angle shots for *hands+objects*.

3.4.3 Camera Movements. The majority (70.59%) of the shots from our sampled videos were shot with static cameras. About 15% of the shots had shaky, abrupt camera movements, which were most detrimental to their visual quality. Only a small proportion (8.13%) of shots applied stable camera movements, such as pan (rotating the camera around its own vertical axis), truck (moving the camera along a horizontal axis perpendicular to the direction of the lens), orbit (moving in an arc while focusing on the same subject), and dolly (moving towards or away from the subject) [7]. This is likely because most instructors do not have access to a dedicated cameraperson.

3.4.4 Summary. Our exploration of camera use in how-to videos helped us to understand the space of camera shots that Stargazer would need to support—what they focus on (the subjects), as well as the range of parameters that would need to be modifiable/used by producers. We next describe the interaction design of Stargazer, which makes use of this vocabulary of shot types, and discuss how Stargazer gives a producer the ability to transition between different types of shots *during* a shoot without needing to break from the instruction or pre-defining camera paths.

4 STARGAZER INTERACTION DESIGN

Stargazer helps instructors produce “one-take” how-to videos with a single camera on a highly articulated, 7-degree-of-freedom robot arm. The system tracks subjects and adjusts camera framing and angle dynamically in response to the instructor’s actions and speech. For instance, the instructor can have Stargazer adjust its view to look at each of the tools they will be using during the tutorial—pointing at the tools one after another to have Stargazer pan to show each of them. Finally, they can say to the audience, “Now if you look at how I put A into B from the top”, Stargazer will respond

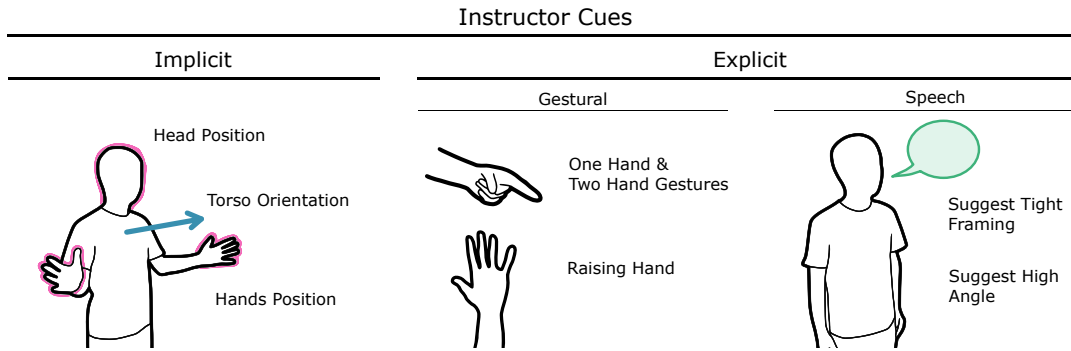


Figure 3: Instructor cues that influence Stargazer behaviors.

by framing their action with the items with a high angle, giving the audience a better view.

Our goal in designing the interaction vocabulary was to identify signals that are smooth, subtle, and could conceivably be a part of the instructor’s dialogue/rapport with the video audience. Our design avoids the need for the instructor to communicate separately to the camera robot independent from the audience; instead, the signals are subtly integrated into the tutorial presentation.

4.1 Capabilities and Needs

Our analysis of how-to videos showed a wide range of video shots and capabilities would be needed by instructors. This provides the building blocks of Stargazer’s functionality. We characterize these based on the vocabulary and coding scheme developed in Section 3.

- *Subject*: The camera needs to be able to capture the subjects common in how-to videos.

Instructor shots primarily frame the instructor for directly addressing the audience—sometimes while presenting objects. (*instructor* and *instructor + objects* from Section 3.)

Action shots capture the instructor’s hands manipulating various tools and objects when performing the demonstrated physical processes. (*hands + objects* from Section 3.)

Object shots focus on one or more objects that the instructor points to (*objects* from Section 3.)

- *Following*: Because the subject may be moving, the robot should automatically locate and follow the subject, maintaining this view within the constraints of the current shot type as well as distance.
- *Camera Framing/Angles*: Instructors should be able to select and transition between normal and tight (zoomed-in) framing, as well as standard and high angles.
- *Camera Movements*: Instructors should also be able to initiate common cinematographic camera movements (e.g. orbit, truck).
- *Feedback*: As with other agents and artifacts that act autonomously [28, 52], the robot needs to provide feedback to the instructor about its current state.

4.2 Interaction Vocabulary

Stargazer responds to instructor signals selected to be subtle (Figure 3). Our goal is to define a vocabulary that ideally fits into the natural flow of the content delivery, without being jarring to the audience and the instructor. The vast majority of the time, Stargazer is understood to be operating autonomously in a reasonable way (i.e., tracking and following the subject with a given framing), and smoothly transitioning between shots to increase legibility. The default camera behavior follows the implicit bodily movements of the instructor and the objects as they are being manipulated. Shifts in the camera position and orientation are driven by signals from the instructors—the robot constantly monitors the instructor’s hands, head, and body postures looking for these signals. Gesture and speech signals from the instructor—ostensibly to the audience—signal to Stargazer changes in high-level camera behaviors, including the subject of shots, camera framing, and camera angles. Our designs select cues that find a balance between subtlety and the chance of false activation.

4.2.1 Body Movement Cues: Hands. The vast majority of how-to video shots are shots of the volume in front of the instructor—where their hands are demonstrating or manipulating an object; this is Stargazer’s default state. For such action shots, Stargazer locates both hands of the instructor and tries to: (1) position both near the center of the frame, and (2) identify a proper distance to capture the hands that accommodate for recent hand movement (Figure 4 A-D). Stargazer tracks the midpoint of the two hands and models a spherical volume that contains the hand movement for the past 5 seconds. This accounts for situations where the distance between the hands changes rapidly—e.g. when the instructor needs to, for instance, select and use tools that are placed adjacent to the object being manipulated. The camera is positioned such that it looks at the midpoint of the two hands and at a distance where the sphere covers one-third of the width of the camera frame.

4.2.2 Body Movement Cues: Head. In instructor shots, the face of the instructor is centered horizontally in the frame, and the camera is oriented perpendicular to the instructor’s torso (Figure 4 E and F). Following the rule of thirds from cinematography principles [7], Stargazer positions the camera such that the instructor’s eyes are at one-third from the top of the camera frame and works to maintain this framing and orientation.

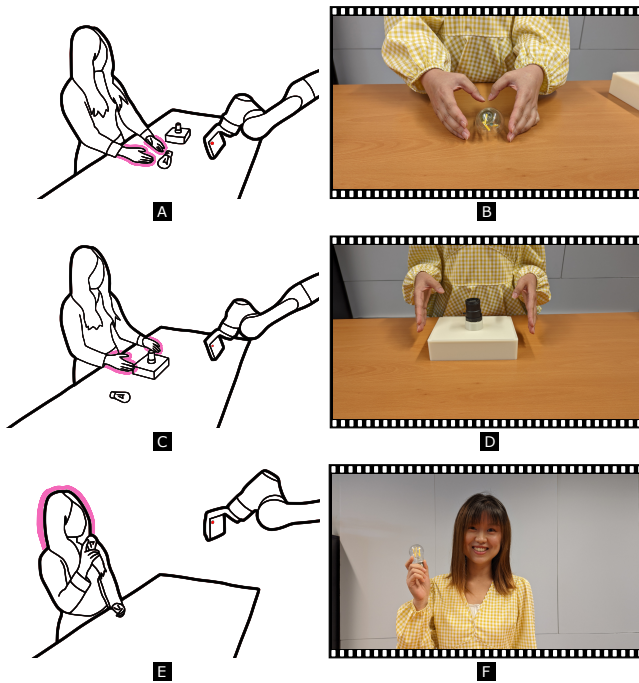


Figure 4: Body movement cues that Stargazer reacts to. (A and C) In action shots, Stargazer follows the hands of the instructor and adjusts its distance to the hands. (B and D) show what the camera captures in A and C. (E and F) In instructor shots, Stargazer follows the head position of the instructor.

4.2.3 Body Movement Cues: Torso Orientation. In instructor and action shots, Stargazer points its camera perpendicularly to the line connecting the instructor’s two shoulders to increase subject visibility. The camera robot reorients itself if an instructor’s torso orientation changes (Figure 5).

4.2.4 Gestural Cues: Single-Hand Pointing. Instructors can direct the attention of the audience by pointing with their index finger at an object—a common communicative action in both everyday life [19] and instructional settings [35]. Stargazer interprets this action as a transition to an object shot by tracking a location that is along the instructor’s pointing direction, close to the fingertip (Figure 6). Stargazer will continue to follow the instructor’s hand

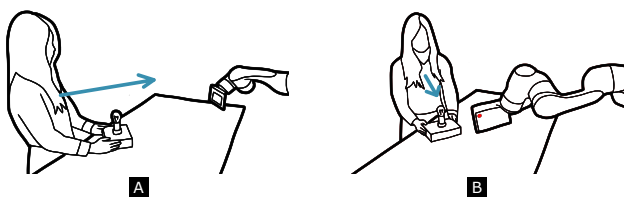


Figure 5: In instructor shots and action shots, Stargazer adjusts its camera orientation and looks perpendicular toward the line connecting the instructor’s shoulders.

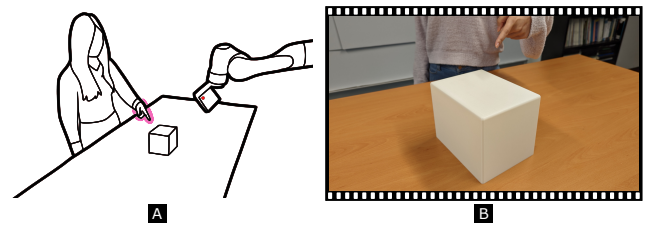


Figure 6: Stargazer transitions to an object shot by following the instructor’s pointing hand.

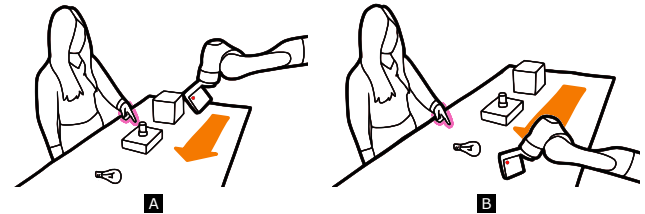


Figure 7: The instructor can move their finger in a horizontal line to trigger a camera “truck” effect (i.e., camera moving horizontally along an axis perpendicular to the lens’ direction)

while they point (this allows for them to point along a trajectory, for example). Once the instructor stops pointing, Stargazer returns to capturing action shots, i.e following the hands.

The instructor can also move their index finger rapidly in a horizontal line while performing the pointing gesture. Stargazer interprets this as a signal to perform a linear tracking effect (a “truck” movement). Its camera follows the finger’s position while being perpendicular to the finger’s trace (Figure 7).

4.2.5 Gestural Cues: Raising Hand. Raising a hand and waving is a familiar gesture for greeting and mediating attention [33], for example, to open up a conversation. Stargazer interprets this hand wave signal as a signal to transition to an instructor shot (Figure 8). Stargazer specifically recognizes a hand above the shoulder for a short period (1s).

4.2.6 Gestural Cues: Two-Hand Pointing. The instructor can initiate a camera orbit movement with one finger pointing from each hand as if they are presenting an object to the audience (Figure 9). Stargazer interprets this gesture as a signal to orbiting around the

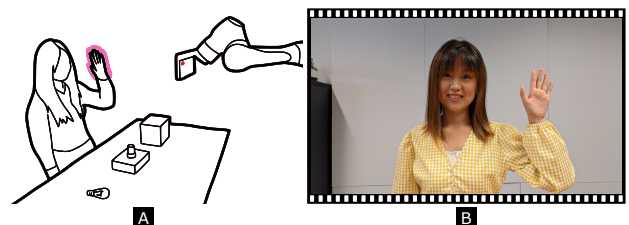


Figure 8: The instructor raises their hand to signal Stargazer to transition to an instructor shot.

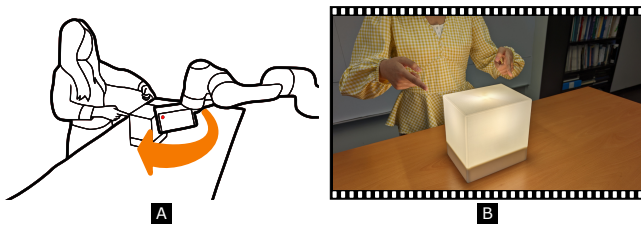


Figure 9: The instructor performs a two-hand pointing gesture to make the camera orbit around the midpoint of the two fingertips.

midpoint of the two fingertips, giving an extended shot while the camera moves in an arc.

4.2.7 Speech Cues: Verbally Suggesting Tight Framing and High-Angle. During in-person instructional settings, an instructor might suggest to a learner to lean in (e.g., “If you take a closer look, ...”) or to watch from a different perspective (e.g. “You can see from the top ...”). Such speech acts are necessary for motor skill learning [13], since they direct the audience to attend to something specific. Stargazer interprets certain pre-defined speech signals from the instructor as suggestions about camera framing and angle when the dialogue seems to make these recommendations to the audience. We implemented two specific speech-triggered behaviors.

Stargazer leverages a state-of-art large language model to infer the intent of the instructor’s speech without enforcing strict templates. The instructor can control zoom-in by saying sentences that suggest taking a closer look. For example: “If you look closer, you can see this socket takes a hexagon shape”, or “Pay more attention to how I take the lid off.” (Figure 10 A and B). Similarly, the instructor can control the camera angle (high-angle shot vs. standard-angle shot) with sentences that suggest viewing from a higher position. For example: “It is better to look from the top to see how I take the headband off”, or “I want you to take a top-down perspective now so that you see the full model.” (Figure 10 C and D). To reset the camera angle to be standard or framing to be normal, the instructor hides one of their hands behind their body.

Stargazer’s use of speech in this way frees the instructor’s hands, which are often occupied in how-to videos. This is markedly different from our use of gestures for shot transitions involving a shift of subject. Those transitions typically occur before/after a complete step when an instructor can choose to empty their hands.

4.3 Feedback to Instructors

Prior research suggests the benefits of providing feedback about an autonomous agent’s states to people who interact with it [28, 52]. Our early pilot tests also show that instructors need to know what the camera “sees” and be able to understand Stargazer’s state and near-future actions. Stargazer provides this feedback with a display attached to the camera mount, which faces the instructor (Figure 11). The display shows a real-time camera preview, and additionally, an icon for the current shot type, and two indicators showing whether a pointing gesture is detected on the left and right hands.

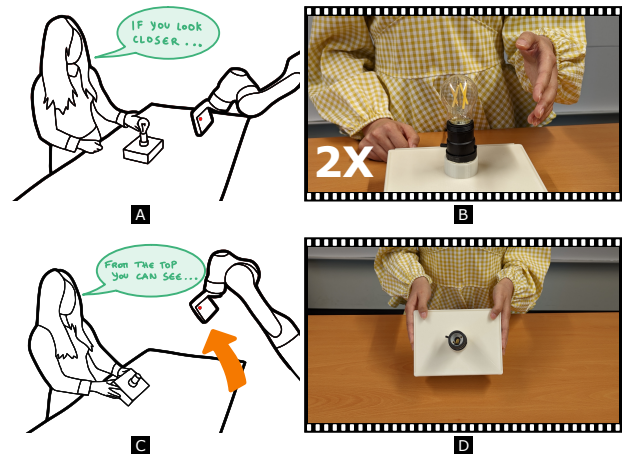


Figure 10: The instructor can (A and B) verbally suggest taking a closer look to make the camera zoom in by 2X, and (C and D) verbally suggest looking from a higher position to take a high-angle shot.

5 IMPLEMENTATION

The Stargazer prototype consists of three main parts: the robot arm, the camera, and the sensors (Figure 12). The robot arm is a seven-degree-of-freedom Franka Emika Panda, controlled by a collection of Robot Operating System (ROS) nodes running on a Linux workstation (the robot workstation). The sensors include a Kinect v2 depth camera, an RGB webcam, and a wireless microphone for sensing the instructor’s skeleton, pointing gestures, and speech. All sensors connect to a Windows workstation (the sensor workstation), which processes sensor data and sends them to the robot workstation through a wired local network in UDP. The camera and the camera monitor are both attached to the robot through a custom mount. The software and hardware design of our prototype is open-sourced online³.

5.1 Camera Hardware and Software

We use the back-facing camera of an Android mobile phone (Google Pixel 6 Pro) as the camera of Stargazer. The phone runs a custom software application to record video and communicate with the robot workstation through a wireless UDP connection. The application shows the preview of the phone’s camera feed and several widgets showing the current shot type and gesture detection results, as described in Sec. 4.3. The robot workstation sends messages to this application to control the camera’s zoom level and update the status widgets. The camera monitor mirrors the screen of the phone to provide feedback to the instructor (more details in Sec 5.1.1). The phone is inserted into a 3D-printed mount (Figure 11), which is attached to the end of the robot arm as a custom end-effector.

5.1.1 Camera Monitor. An iPod Touch is mounted to the instructor-facing side of the camera monitor. It displays the preview of the mobile phone’s camera feed and robot status, mirroring the phone’s screen through a local wireless network. We

³<https://github.com/jchrisli/stargazer-chi23>

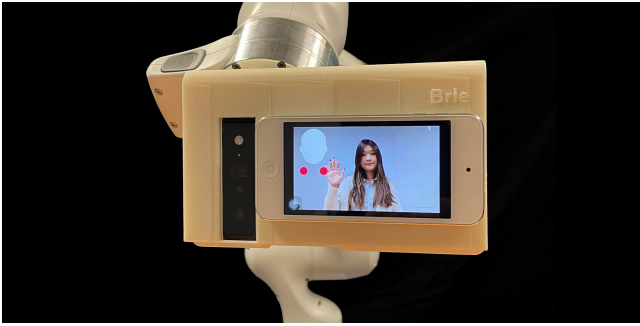


Figure 11: The camera monitor provides feedback to the instructor, showing the camera preview, the current shot type, and whether a pointing gesture is detected. In this photo, the monitor shows a head icon indicating the current shot type is an instructor shot, and two red dots indicating neither hand is pointing.

use the sensor workstation as a relay point for screen mirroring. The sensor workstation first obtains a screencast of the phone through scrapy⁴ and streams the screencast to the iPod Touch via Moonlight⁵.

5.2 Sensing Instructor’s Actions

The Stargazer prototype employs a combination of sensors to detect the body pose, gesture, and speech of the instructor. Future work can explore leveraging the sensing capabilities of the robot’s camera itself.

5.2.1 Body Pose. We point a Kinect v2 depth camera at the instructor and infer the instructor’s body pose using the Kinect SDK for Windows. Stargazer uses the head joint as a proxy for the face, the fingertip joints for the hands, and the vectors from the wrist joints to the corresponding fingertip joints for pointing directions. In object shots, we track a location that is 12cm from the fingertip along the pointing direction. To estimate the spherical volume that contains both hands, we run a rolling average of the distance between the hands for the most recent 5 seconds. For filtering and streaming the data, we used the Creepy Tracker toolkit [54].

5.2.2 Hand Gesture. A 720p RGB webcam is aimed at the center of the workbench to detect the instructor’s pointing gestures. We locate the 2D keypoint positions of the instructor’s hands using a deep-learning-based detector [34] and classify whether the hand is performing a pointing gesture (index finger extending forward while middle, ring, and pinky curled up) through a custom model. The 21 key point positions from each hand have been normalized before classification. The gesture detection model is a four-layer fully-connected neural network implemented with Keras, trained with data collected from the research team. To reduce ambiguity in classification, we only classify the current gesture as pointing if the model has at least 85% confidence. We further apply low-pass filtering to remove sporadic false positives.

⁴<https://github.com/Genymobile/scrcpy>

⁵<https://moonlight-stream.org/>

5.2.3 Speech Understanding. We record the instructor’s speech with a wireless microphone clipped to the instructor and send it to a speech recognition API (Microsoft Azure Speech-to-Text⁶). The transcribed text, along with a custom prompt⁷, is then sent to a large language model (GPT-3 [8]), which labels the intent (i.e., tighter framing/high angle/normal) of the instructor.

5.3 Robot Control

The Stargazer prototype uses ROS to communicate with the robot arm and control its motion. The robot workstation runs a ROS node that computes the next target camera pose based on sensor inputs. We describe the algorithm for planning the next camera poses in Section 5.4. Given a target camera pose, we get the desired Cartesian linear and angular velocities with a PID controller. A real-time servoing library, MoveIt Servo, converts the Cartesian velocities to joint velocities. Finally, we set the joint velocities through the Franka ROS interface⁸. The next camera pose is calculated at a frequency of 5 Hz. The PID control loop runs at 100 Hz.

We monitor the robot’s status to determine if it is close to kinematic singularities or joint limits which would render it unable to move properly. If so, the control program pauses real-time velocity control and recovers the robot from the error state by returning to a safe neutral position.

5.4 Robot Motion Planning

While the camera pose has six degrees of freedom, Stargazer’s camera always aims to look at a known location (e.g. the instructor’s head) and keep its image plane’s horizontal axis parallel with the ground. This reduces our motion planning problem from \mathbb{R}^6 to \mathbb{R}^3 . We aim to plan the next target camera position $\mathbf{x}_{t+1} \in \mathbb{R}^3$ for capturing the subject (e.g., face, both hands) at position $\mathbf{x}_s \in \mathbb{R}^3$ under a number of constraints, such as motion smoothness and distance to the subject, at time step t . Except for planning the orbit paths, we formulate the next position selection as an optimization problem, following prior work in autonomous drone cinematography [41]. We calculate the orbit path directly as a series of waypoints, as described in Section 5.4.2.

5.4.1 Motion Planning Cost Function. The cost function J consists of the weighted sum of the following cost values.

Motion Smoothness: To achieve smooth camera motion, we penalize large displacement in the camera position with the cost $c_{ms} = (\mathbf{x}_{t+1} - \mathbf{x}_t)^2$.

Desired Distance: We first compute the desired distance d to film the subject based on the current type of shot. For action shots, the subject should cover one-third of the camera frame’s width. Therefore, for a subject with radius r and a camera with field-of-view α , the desired distance $d = 2r \tan(\alpha/2)/3$. For instructor shots and object shots, we empirically set d to be 0.20m and 0.12m, respectively. Given d , the desired-distance cost is $c_{dd} = (\|\mathbf{x}_{t+1} - \mathbf{x}_s\| - d)^2$. We do not enforce distance cost for high-angle shots as it tends to drive the robot into kinematic singularities.

Pitch: For instructor shots and shots with a high-angle setup, we encourage a camera pitch value close to $\pi/2$ and 0, respectively.

⁶<https://azure.microsoft.com/en-us/services/cognitive-services/speech-to-text/>

⁷We attach the prompt we used in the supplemental materials.

⁸<https://frankaemika.github.io/docs/index.html>

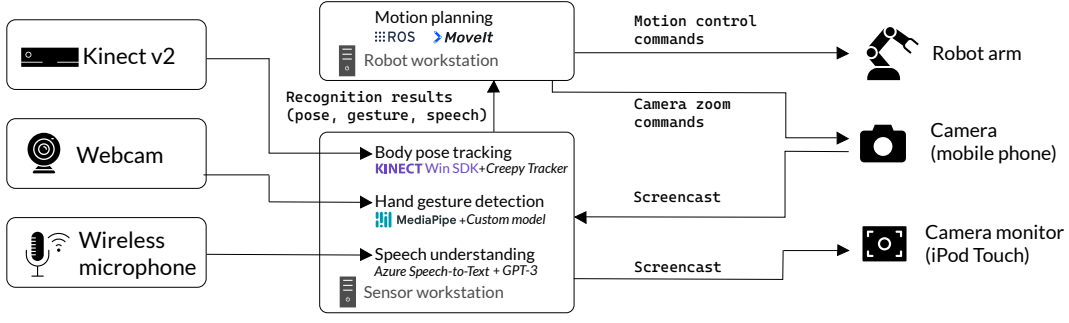


Figure 12: An overview of the components of the Stargazer prototype.

We calculate a pitch cost for the desired pitch value β as $c_p = ((\mathbf{x}_s - \mathbf{x}_{t+1}) / \|\mathbf{x}_s - \mathbf{x}_{t+1}\| \cdot \mathbf{v}_g - \arccos(\pi/2 - \beta))^2$, where \mathbf{v}_g is the gravity direction $(0, 0, -1)$ (see Figure 13A).

Orientation: We add a camera orientation term c_o in two conditions. For face and action shots, we encourage the camera to look perpendicularly at the line linking the instructor’s two shoulders. During truck movements, we guide the camera to point perpendicularly at the finger’s movement direction. Since we already have a cost term for controlling camera pitch, we only need to penalize the difference between the camera orientation vector of the next step and the target camera orientation vector in the ground plane (Figure 13B). We refer our readers to Appendix. A.1 for the detailed derivation of c_o .

In summary, the cost function for selecting the next camera position is

$$J = \omega_{sm}c_{sm} + \omega_{dd}q_{dd}c_{dd} + \omega_pq_p c_p + \omega_oq_o c_o$$

where $\omega_{sm}, \omega_{dd}, \omega_p, \omega_o$ are weights for the cost terms. If the current shot is not a high-angle shot, $q_{dd} = 1$ otherwise 0. If the current camera angle is set to high-angle or the current shot is an instructor shot, we set $q_p = 1$, otherwise 0. If the current shot is a hand or instructor shot, or the camera is performing a truck movement, $q_o = 1$ otherwise 0. For weights, we empirically set $\omega_{sm} = 1.0$, $\omega_{dd} = 0.2$, $\omega_p = 1.0$, $\omega_o = 0.5$ to achieve a balance between responsiveness and smooth tracking. To simplify the bounds on the variables, we re-parameterize \mathbf{x}_{t+1} as $(\theta_{t+1}, \psi_{t+1}, R_{t+1})$ in a polar coordinate system. We solve this nonlinear optimization problem

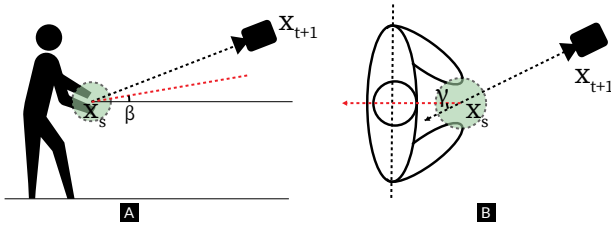


Figure 13: (A) The cost for the difference between the camera pitch angle of the next camera position \mathbf{x}_{t+1} and the desired pitch angle (in red) β . (B) The cost for the desired camera orientation (in red) and the orientation of the next camera position \mathbf{x}_{t+1} . γ indicates the difference.

with the SLSQP solver in SciPy. We refer our reader to Appendix. A.2 for the details of the polar coordinate system used and the bounds on the variables.

5.4.2 Orbit Paths. Given an orbit center, the Stargazer camera always travels along an arc of $\pi/4$ and looks at the orbit center from a distance of 60cm with a pitch angle of $\pi/6$.

5.4.3 Implementation Details. The robot workstation runs Ubuntu 20.04 and ROS 1 Neotic with an Intel Xeon W2295 CPU and 503 GB memory. The sensor workstation runs Windows 11 with an Intel i7 8750 CPU and 16 GB memory. We implemented one performance-critical ROS node, which calculates end-effector velocities, in C++, and the other nodes in Python. The speech detection and labeling program were implemented in C#. The programs for other sensors were implemented in Python.

6 EVALUATION

We conducted a preliminary user evaluation where six instructors created video tutorials demonstrating distinct physical skills. The goal of the evaluation was to gain an initial understanding of Stargazer’s ability to support the filming of how-to videos and the instructor’s perception of working with Stargazer.

6.1 Study Design and Participants

We invited six physical skill instructors (four female, two male) to create video tutorials for skills that they have expertise in. Three participants (P1, P5, and P6) were artists who practiced the craft explained in their tutorials professionally. The other three participants were experienced hobbyists with more than a year of experience in the skill they taught. Two participants (P2 and P5) had experience teaching physical skills through online videos or live streaming. P5 and P6 were professional filmmakers. See Table. 1 for more details on the participants. Participants were compensated for their time.

The study protocol has been approved by the ethics review board of our institutions.

6.2 Task and Procedure

We asked the participants to freely demonstrate their physical skills without constraints. The only limitations were the scale (the task was demonstrated on a workbench) and the length of the video instruction (around 5 minutes per session). Participants brought their own materials. Figure 14 shows a participant filming a video

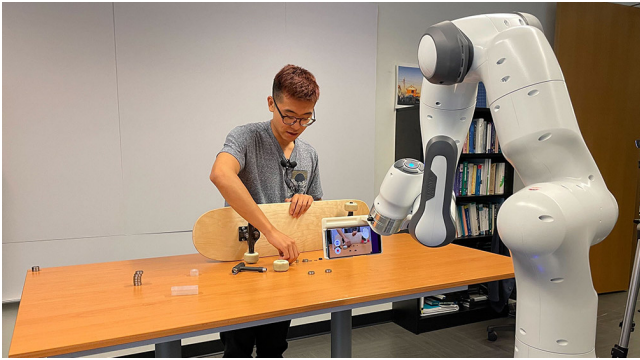


Figure 14: Participant recording a how-to video on skateboard maintenance.

with Stargazer. Each session lasted around 90 minutes and consisted of three phases:

Training. After filling out a demographics survey, participants went through a training session where they learned about the features of Stargazer, including the types of shots it offers and the interactions it supports for changing camera behaviors. They experimented with all the features and recreated a short Lego assembly tutorial video, which required using each feature at least once.

Rehearsal and Review. After participants successfully recreated the Lego tutorial, they filmed a rehearsal version of their own physical skill tutorial with Stargazer. The experimenter did not provide any guidance or feedback during filming. The participants then reviewed the rehearsal version with the experimenter, using think-aloud to remark on the video and reflect on which parts could be improved. We included the rehearsal stage for participants to practice delivering their instruction.

Filming and Review. Participants then created a second take of the tutorial video, incorporating changes in content and delivery they had planned after reviewing the rehearsal version. Again, participants used think-aloud to review the outcome and comment on the

ID	Background	Skill Taught
P1 (Pro)	Artist in digital fabrication.	Photogrammetry and 3D printing.
P2 (EH)	Skateboarding influencer, running an Instagram channel teaching skateboarding skills.	Skateboard maintenance.
P3 (EH)	Virtual reality researcher.	Head-mounted display set-up.
P4 (EH)	Crafting hobbyist with art training.	Clay model making.
P5 (Pro)	Filmmaker, teaching stop-motion filmmaking to college students and the general public.	Stop-motion puppet making.
P6 (Pro)	Digital artist and filmmaker.	Interactive sculpture making.

Table 1: Evaluation Participants (Pro - Professional; EH - Experienced Hobbyist)

quality of the video and their experience when filming it. Following the review phase, the experimenter conducted a semi-structured interview with participants, focusing on alternative ways to film the videos, how the robot could fit into their current tutorial creation workflow, and comparing Stargazer with existing video capturing solutions for real-time transitions between multiple camera sources, such as Open Broadcast Studio (OBS). Finally, participants answered a questionnaire consisting of 7-point Likert scale questions about their perception of the robot’s behaviors, the interactions, and the video they produced. We recorded robot motion and user input logs for preliminary performance analysis. As a first step in prototyping interactive camera robots for filming how-to videos, our data collection focused on the perspectives of instructors to get their firsthand account of how Stargazer’s interaction design might affect instruction delivery and video quality. Future work could instead study the perspectives of external experts and learners on produced videos.

6.3 Setup and Apparatus

We ran the study in a controlled lab environment. The workbench where the participants demonstrated their skills was 1.6 meters x 0.8 meters, placed 1.2 meters from the base of the robot. We chose this physical setup and placed constraints in motion planning so that the robot was distant enough to avoid any collision between the robot and the instructor. One researcher guided participants through the experiment and conducted the interview. The setup of the robot is as described in Section 5.

6.4 Results

In this section, we describe the video results created by participants, their perception of interacting with Stargazer, and preliminary technical performance measures.

6.4.1 Training. All participants were able to successfully use all the features of Stargazer while filming the Lego assembly tutorial in the training session. Participants picked up gesture-based interactions very quickly but it took them 2-3 trials to learn to add speech commands into instruction content.

6.4.2 Video Content. All participants were able to complete their tutorial videos after the first (rehearsal) or second take. Two participants (P4 and P6) were satisfied with the results of their first takes and stopped filming. The remaining participants did a second take. Our following analysis of the video content is based on the final versions of the video results.

Figure 15 illustrates keyframes and timeline summaries of each of the videos created by participants. The lengths of the final videos ranged from 245 seconds (4m 5s) to 451 seconds (7m 31s) ($M=320s$, $SD=69s$). The skills in each video varied significantly in terms of the artifacts and processes required, and participants imprinted their personal style in the delivery of the content. Overall, 16.1% of the total video duration featured instructor shots, 78.4% was action shots, and 5.5% was object shots. Participants incorporated a rich variety of shots by leveraging different combinations of subjects, camera framing, and camera angles (see Figure 15). On average, each video had 7.7 times of subject changes (e.g., from an action shot to an object shot, $SD=2.6$), 5.7 framing changes (e.g., zoom-in,

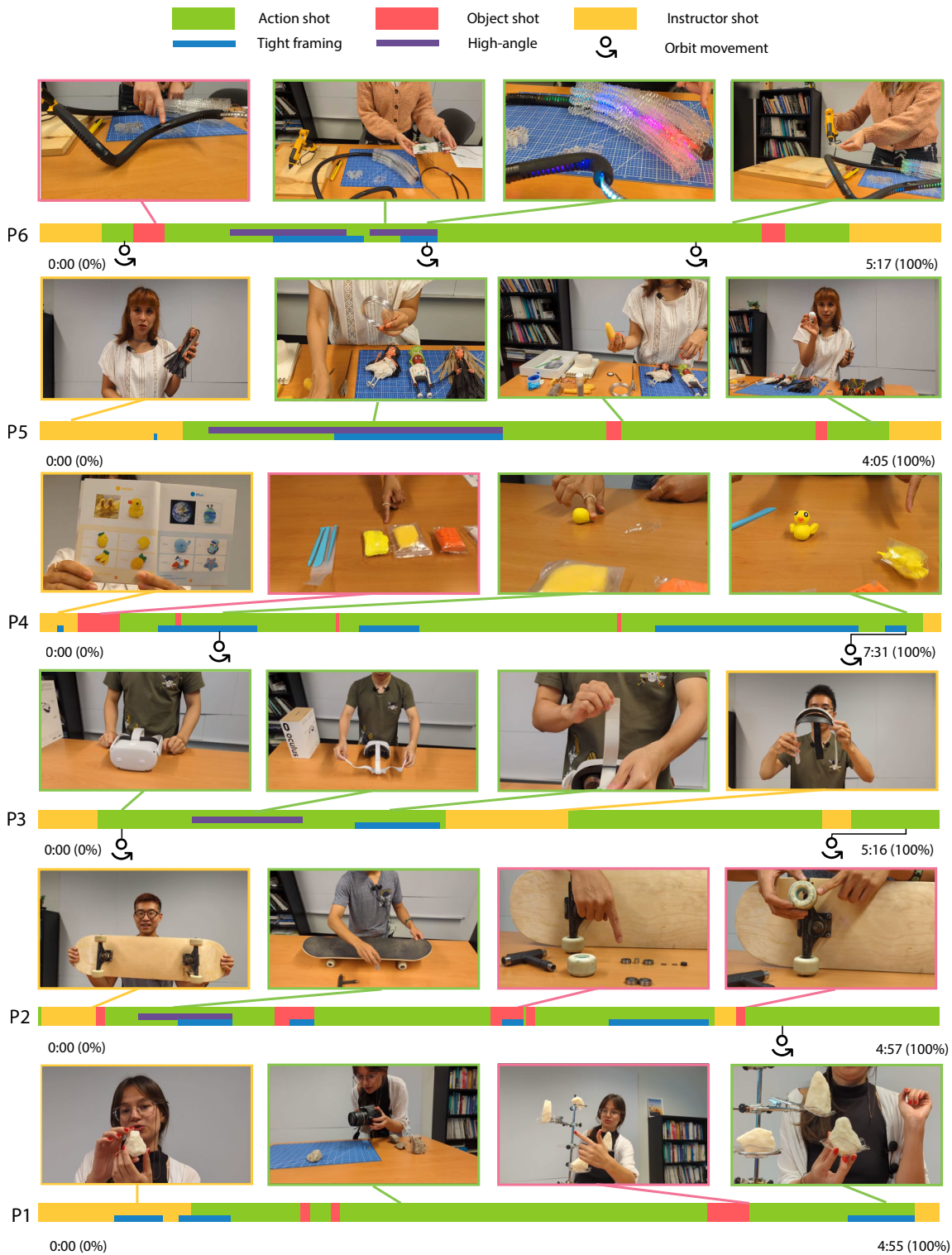


Figure 15: Timelines with the videos’ participants filmed with Stargazer. The timelines are color-coded for the shot type, camera angles, camera framing, and camera movements. The representative frames for each video are above the timelines.

SD=2.9), and 1.7 angle changes (e.g., from standard to high angle, SD=1.5). Participants were also able to supplement their videos with camera movements, albeit to varying extents. They used orbit movements 1.2 times per video on average (SD=1.2). There was only one instance of a participant using the truck movement. We attach the participant-produced videos in our supplemental materials.

6.4.3 Video Quality. Participants reviewed the tutorial videos they produced and remarked on the quality of the results. All participants reported they were able to successfully deliver the instruction content that they had planned in the produced videos (M=6.5, IQR=0.75). Additionally, they found the videos informative (M=6.3, IQR=1.0), with good visual quality (M=6.5, IQR=0.75) and cinematographic quality (M=6.0, IQR=1.5). One participant (P6), who is a professional filmmaker, highlighted the visual characteristics of “one-take” videos, which Stargazer films, “*there is interesting theory attached to it (one-take films) about continuity and slowness that allows the viewer to be fully present.*”. Indeed, professionals have long valued cinematic continuity in film editing and research in cognitive science posits that continuity helps with guiding viewers’ visual attention [53]. An interesting question for the fields of HCI and educational psychology is how cinematic continuity in learning materials may affect physical skill learning.

6.4.4 Participants’ Overall Perception. Participants found Stargazer useful for creating tutorial video content (M=6.3, IQR=0.75). The professional artist participants (P1, P5, P6) stated that they often need to document their work processes through video, “*we ask our colleagues to help (filming) but they are not always available*” (P1). When comparing Stargazer to their current documentation workflow, they appreciated that it could alleviate the burden of setting up cameras and post-editing, “*do my thing and it (filming) just happens*” (P5).

6.4.5 Robot Behaviors and Instructor Control. We conducted an in-depth analysis into how the behaviors of Stargazer and the associated interactions worked for participants, based on a synthesis of interview data, questionnaire responses, and their think-aloud comments while reviewing the captured videos.

Shot types. All participants found the three types of shots—action, instructor, and object—sufficient for their purposes when creating the tutorials (M=6.0, IQR=1.5). As expected, participants mostly used instructor shots to introduce themselves, the task, and certain objects. They used action shots for illustrating the steps and used object shots to highlight objects and details on objects. However, we also observed that participants were able to flexibly adapt the robot’s behaviors for their in-situ filming needs and blurred the boundaries between the anticipated purposes of the shots. For example, in her tutorial on stop-motion puppets, P5 often described a component while holding it close to her chest. This hand position leads to the camera capturing both her face and hands while the robot is still in the mode of filming action shots. She explained that she guided the robot’s camera into this position as she wanted to talk to the audience face-to-face more often.

Camera framing and angles. Most participants (5/6) reported that they were able to film the tutorial content with the framing and angle they desired (M=5.3, IQR=2.5 for framing, M=5.1, IQR=1.0

for angles). All participants see clear utility in the ability to adjust framing (M=6.6, IQR=0.75) and angles (M=6.5, IQR=1.0) using speech.

Participants mostly used normal (medium for instructor shots, close-up for action and object shots) framing (78.9% of the total video duration), but switched to tighter framing to highlight details. Control of framing especially benefited tasks that involved more actions and objects at a fine scale. P4 applied tighter framing for 43% of her full video to ensure that individual parts of the clay model, including smaller components like facial features, were clearly visible to the audience (e.g. the last keyframe on P4’s timeline in Figure 15). Participants used high-angle shots to show an overview of the workbench (e.g., P2’s second keyframe on Figure 15) or to take a different, visually interesting perspective of the same object (e.g., P5, P6, the second keyframes on both of their timelines in Figure 15). After showing how individual components constitute a sculpture, P6 made the robot look downward to show the sculpture’s complete structure.

Blending Camera Control into Instruction. Overall, participants found that they could blend camera controls fluidly into instructional activities. One participant (P5) drew on her experience teaching live remote art classes via a video recording tool (Open Broadcaster Software, or OBS), which supports real-time switch between video sources, and stated that Stargazer’s controls did not force her to “*stop my work and reach for that box (a keypad for changing camera sources in OBS) every time*”.

Four participants explicitly commented on the ease of using the pointing gesture to initiate object shots, finding that it naturally fit into their actions without being disruptive to instruction (M=6.7, IQR=0.75), as P6 described, “*do not need to think about it*”. P4 liked that she could quickly shift the camera between a tool and the work at hand through pointing. However, P1 found that the transient nature of the pointing gesture somewhat limited her freedom to construct an object shot. To make the camera lock onto a particular object while she was fetching other tools, she had to keep pointing at the object with one hand and grab the tools with her other hand. She suggested that the robot should “*keep focusing on an object unless I tell it to stop*”.

Participants reported that raising their hand to trigger instructor shots did not disrupt their flow but was less natural than pointing (M=5.2, IQR=2.25). P3 explained “*it (raising a hand) feels very natural when you have something to show in you hand but not so much if you just want to quickly talk to your audience*”. P6 suggested the potential of gaze, “*I’d like it to turn to my face if I look at it, and go back to my hands if I look away*”.

Five out of the six participants found controlling camera framing and angles with speech did not disrupt their instruction delivery (M=5.5, IQR=1.75). P2 considered it his favorite feature, as “*I can call it to zoom in anytime even if both of my hands are occupied...Being able to show the details makes me want to explain more because they (the audience) can see it.*” Participants liked the flexibility that Stargazer offered in choosing speech commands; however, they also expressed that the uncertainty in “*what the robot actually understands*” (P1) could make them overthink what to say.

6.4.6 Preliminary Performance Measures. We report objective measures for several aspects of the Stargazer prototype based on data

logs and human labels of study session recordings. Note these measures were only meant to obtain an initial understanding of the technical approaches we used. We included recordings of the rehearsal and the final takes (if available) in the analysis. The total duration of the video analyzed was 71 minutes and 22 seconds.

Pointing Gestures. We compared the human labels for participants' gestures and detection result logs. The precision for pointing detection was 91.2%, and the recall was 86.6%.

Speech Commands We calculated the metrics for speech commands, labeling on a per-sentence basis (i.e., "you can take a close view here" as one sentence). There were in total 41 sentences labeled as suggesting "tight framing" by humans and 40 by Stargazer. The precision for the label was 90.0%, and the recall was 87.8%. There were in total 18 sentences labeled as suggesting "high-angle" by humans and 17 by Stargazer. The precision for the label was 94.1%, and the recall was 88.9%. We also calculated the delay between participants finishing a sentence with a particular intention (e.g. tight framing) and the time the robot received the corresponding state change command. The average delay was 2.4s (SD=2.3s). The large variance might be due to the variance in speech recognition results.

The goal of the Stargazer prototype is to demonstrate and evaluate the concept of directing camera operations by following instructor cues. We anticipate that with the advance in machine learning and sensing technologies, the performance of the techniques we applied will continue to improve.

7 DISCUSSION

Drawing on evaluation results and observations, in this section, we discuss the challenges in interpreting instructor intent for camera robots, routes to support other content creation workflows, and finally, limitations and future work.

7.1 Interpreting Instructor Intent

One goal of our interaction designs for Stargazer is to identify and take advantage of instructor cues that clearly express their intent in camera configuration. While our design received encouraging feedback from the evaluation, we also noted two general challenges for future camera robots that interpret instructors' or other actors' intent to assist in video creation.

The first challenge is to understand instructor intent that is not well articulated. Although participants generally did not find incorporating camera control cues disrupted instruction, they still have to "*actively remember when to use what*" (P3). P3 expected future robots to be more autonomous, "*(content creators) can just do their things and not think about there is a camera*" (P3). P6 suggested that future robots could introduce more variance in camera motion that fits the current scene without instructors' commands. A camera robot that can produce quality footage without deliberate human input should understand what is to be captured and how to capture them. However, the instructor may not want to specify them or do not know how to articulate them clearly. Such intents can be difficult to glean from instructor cues, as cues including gestures and speech are often ambiguous when taken out of their semantic (e.g., what is the current task) and temporal (e.g., the instructor's actions so far) contexts [18]. One exciting opportunity for this challenge may

be a data-driven approach that infers intent based on the semantic content of tutorials in the format of scripts, storyboards, or a history of the objects that the instructor has interacted with so far.

The other challenge is to build personalized robot behaviors that adapt to an instructor's preferences and style. Content creators trying to explore the frontier may seek camera shots that are "out-of-distribution" rather than the most common. For these explorers, programming-by-demonstration or other more explicit robot programming methods may help complement data-driven or heuristics-based approaches.

7.2 Support Other Content Creation Workflows

Stargazer currently captures videos with a single continuous take. While one of the participants (P6), a filmmaker, highlighted the aesthetic characteristics of one-shot videos, this is not the only format that how-to video creators adopt. For example, P2, who runs a skateboarding social media channel, discussed the opportunity to capture skateboard drills with Stargazer and pointed out that most of his videos are heavily edited short clips for mobile consumption. He must also perform and film the same drill several times until he gets satisfactory results.

Many content creators edit raw footage for conciseness, clarity, and visual quality. For example, some content creators might want to remove parts of the footage where the robot transitions from tracking one subject to another. Future camera robots could assist with their workflows using the robots' data logs. Algorithms could automatically remove these parts of the videos in post-processing based on the robots' state history. Alternatively, the system could cut to a second, fixed camera's feed while the robot is in transition to avoid delay. For instructors who need multiple takes of the same process, future systems could provide interfaces for instructors to plan the robot's motion in future takes based on the recording of the first take.

7.3 Limitations and Future Work

The robot arm's kinematic constraints, including its fixed base and finite reach, place limits on the range of possible camera positions. Stargazer's range of possible camera positions is sufficient for tabletop-scale activities, but not for tasks involving multiple distant locations in a room. These constraints also restrict the use of certain camera angles (e.g. point-of-view, top-down) and camera movements (e.g. 180° orbit). In future work, we are interested in the potential of camera drones [50] and mobile manipulators [3] to assist with filming tasks that take place in larger environments from a wider variety of angles.

The cues that Stargazer reacts to do not include all the rich nuances in instructors' intents. For example, some participants attempted to trigger object shots with deictic gestures such as giving or showing objects to the camera, which were not part of the cues that Stargazer recognizes. We observed that most participants went through a shift of their mental models about working with Stargazer, during the training phases or the first takes. They began to consciously take into account what the robot could understand when delivering the tutorials, as P5 put it, "*I started to see this is a performance of me and the robot together.*" Future research could

investigate methods to detect diverse and subtle intents, possibly by combining multi-modal signals including gaze, posture, and speech.

Our focus in the video analysis and interaction design for Stargazer was on understanding and producing videos that emphasize instructional content. Other types of physical-skill-focused videos, such as creative livestreams, could include significant non-instructional components like socializing and performing [17] and could be several hours long. We plan to further study the content in these videos and their creators' camera control needs to identify additional cues and robot behaviors for Stargazer to better support non-instructional content.

The dataset for our video analysis used the number of views as a proxy for sufficient video quality, but we did not directly assess the quality of these videos. Thus, our data includes common instructor practices, but not necessarily the best practices. Datasets with comprehensive quality metrics could be constructed in the future to inform guidelines for tutorial video filming.

As a research prototype, the current Stargazer prototype relies on an expensive, general-purpose robot arm and a suite of external sensors. As robot-assisted filming becomes more prevalent, there could be demand for more affordable, easier-to-use, and lighter-weight specialized robots for content capture. Future work could explore the technical and interaction design of such robots, and their implications for content creation on a broader scale. An alternative approach is to apply a subset of Stargazer's camera control interactions to ready-to-use technologies, such as camera setups consisting of multiple fixed cameras.

We apply a recovery motion to free the Stargazer robot from configurations close to kinematic singularities [15] or joint limits. While the recovery motion is brief, it compromises the fluid cinematic continuity. Computing fully singularity-free motion is still an active research area in real-time robot motion synthesis [45]. A promising solution to this problem is to relax the constraints on one or more degrees of freedom in motion planning. In the context of robot cinematography, an example of this approach is to ignore the camera-to-subject distance.

While the videos captured by Stargazer were positively received by the instructors themselves, we recognize that participants could display self-evaluation bias towards their own work. Further evaluations could involve external experts on the skill taught and cinematography experts to counter such possible bias. We also note that the design and evaluation of Stargazer have not yet taken the perspectives of the end consumer of how-to videos. Our future work will investigate learners' perceptions of robot-captured videos and compare them to baseline approaches such as videos filmed with fixed cameras. We are also interested in interaction designs for learners to influence the behaviors of camera robots in the context of live how-to tutorials.

8 CONCLUSIONS

In this paper, we proposed Stargazer, a new approach to capturing dynamic how-to video content with a camera robot that reacts to subtle cues from the instructor. Stargazer's camera automatically follows the hands or face of the instructor, and transitions between these subjects of interest as the instructor suggests. The instructor can use gestures and speech that are ostensibly addressed to the

audience to signal to Stargazer to change the subject of interest it follows, its camera framing, and camera angle. Our interaction design for Stargazer was informed by an analysis of 50 online how-to videos. A user study with six participants, including two professional filmmakers, showed that physical skill instructors can successfully create how-to video content with Stargazer and were satisfied with the produced videos' quality. They also did not find the interaction disruptive to instruction delivery. Stargazer explores the potential of robot-assisted video filming for how-to videos, and we hope it can inspire future research in human-robot collaborative content creation.

ACKNOWLEDGMENTS

Authors would like to thank the support from the participants, the anonymous reviewers, and NSERC grants RGPIN-2017-06415, DGDND-2017-00093, IRCPJ 545100-18, SMFSU 549149-2020, and RGPIN-2017-04883.

REFERENCES

- [1] 2022. Bolt: High-speed small cinema robot: Affordable Robotic Solutions. <https://www.mrmoco.com/motion-control/bolt/>
- [2] Henny Admoni and Siddhartha Srinivasa. 2016. Predicting user intent through eye gaze for shared autonomy. In *2016 AAAI Fall Symposium Series*.
- [3] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, et al. 2022. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691* (2022).
- [4] Reuben M Aronson and Henny Admoni. 2020. Eye gaze for assistive manipulation. In *Companion of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*. 552–554.
- [5] Reuben M Aronson, Nadia Almutlak, and Henny Admoni. 2021. Inferring Goals with Gaze during Teleoperated Manipulation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 7307–7314.
- [6] Rogerio Bonatti, Wenshan Wang, Cherie Ho, Aayush Ahuja, Mirko Gschwindt, Efe Camci, Erdal Kayacan, Sanjiban Choudhury, and Sebastian Scherer. 2020. Autonomous aerial cinematography in unstructured environments with learned artistic decision-making. *Journal of Field Robotics* 37, 4 (2020), 606–641.
- [7] Blain Brown. 2016. *Cinematography Theory and Practice: Imaagemaking for Cinematographers & Directors*. Routledge.
- [8] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.
- [9] William Cerbin. 2018. Improving student learning from lectures. *Scholarship of Teaching and Learning in Psychology* 4, 3 (2018), 151.
- [10] Minsuk Chang, Leonore V. Guillain, Hyeungshik Jung, Vivian M. Hare, Juho Kim, and Maneesh Agrawala. 2018. RecipeScape: An Interactive Tool for Analyzing Cooking Instructions at Scale. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) (*CHI '18*). Association for Computing Machinery, New York, NY, USA, 1–12. <https://doi.org/10.1145/3173574.3174025>
- [11] Minsuk Chang, Mina Huh, and Juho Kim. 2021. RubySlippers: Supporting Content-Based Voice Navigation for How-to Videos. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) (*CHI '21*). Association for Computing Machinery, New York, NY, USA, Article 97, 14 pages. <https://doi.org/10.1145/3411764.3445131>
- [12] Pei-Yu Chi, Joyce Liu, Jason Linder, Mira Dontcheva, Wilmot Li, and Bjoern Hartmann. 2013. DemoCut: Generating Concise Instructional Videos for Physical Demonstrations. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology* (St. Andrews, Scotland, United Kingdom) (*UIST '13*). Association for Computing Machinery, New York, NY, USA, 141–150. <https://doi.org/10.1145/2501988.2502052>
- [13] Gibson Darden and Jane Shimon. 2000. Revisit an "old" technology videotape feedback for motor skill learning end performance. *Strategies* 13, 4 (2000), 17–21.
- [14] Björn B. de Koning, Vincent Hoogerheide, and Jean-Michel Boucheix. 2018. Developments and Trends in Learning with Instructional Video. *Computers in Human Behavior* 89 (2018), 395–398. <https://doi.org/10.1016/j.chb.2018.08.055>
- [15] Peter Donelan. 2010. Kinematic singularities of robot manipulators. In *Advances in Robot Manipulators*. InTechopen.

- [16] Maxwell Forbes, Rajesh PN Rao, Luke Zettlemoyer, and Maya Cakmak. 2015. Robot programming by demonstration with situated spatial language understanding. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014–2020.
- [17] C. Allie Fraser, Joy O. Kim, Alison Thornsberry, Scott Klemmer, and Mira Dontcheva. 2019. Sharing the Studio: How Creative Livestreaming Can Inspire, Educate, and Engage. In *Proceedings of the 2019 on Creativity and Cognition* (San Diego, CA, USA) (C&C '19). Association for Computing Machinery, New York, NY, USA, 144–155. <https://doi.org/10.1145/3325480.3325485>
- [18] Euan Freeman, Stephen Brewster, and Vuokko Lantz. 2016. Do That, There: An Interaction Technique for Addressing In-Air Gesture Systems. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) (CHI '16). Association for Computing Machinery, New York, NY, USA, 2319–2331. <https://doi.org/10.1145/2858036.2858308>
- [19] Kathleen R Gibson, Kathleen Rita Gibson, and Tim Ingold. 1994. *Tools, language and cognition in human evolution*. Cambridge University Press.
- [20] Brian Gleeson, Karon MacLean, Amir Haddadi, Elizabeth Croft, and Javier Alcazar. 2013. Gestures for industry intuitive human-robot communication from human observation. In *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 349–356.
- [21] Pavel Gurevich, Joel Lanir, Benjamin Cohen, and Ran Stone. 2012. TeleAdvisor: A Versatile Augmented Reality Tool for Remote Assistance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Austin, Texas, USA) (CHI '12). Association for Computing Machinery, New York, NY, USA, 619–622. <https://doi.org/10.1145/2207676.2207763>
- [22] D. Brian Hall, Lyn Bartram, and Matthew Brehmer. 2022. Augmented Chironomia for Presenting Data to Remote Audiences. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology (UIST '22)*. Association for Computing Machinery, New York, NY, USA.
- [23] Vincent Hoogerheide, Margot van Wermeskerken, Hilke van Nassau, and Tamara van Gog. 2018. Model-observer similarity and task-appropriateness in learning from video modeling examples: Do model and student gender affect test performance, self-efficacy, and perceived competence? *Computers in Human Behavior* 89 (2018), 457–464.
- [24] Chong Huang, Fei Gao, Jie Pan, Zhenyu Yang, Weihao Qiu, Peng Chen, Xin Yang, Shaojie Shen, and Kwang-Ting Cheng. 2018. ACT: An Autonomous Drone Cinematography System for Action Scenes. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. 7039–7046. <https://doi.org/10.1109/ICRA.2018.8460703>
- [25] Chien-Ming Huang and Bilge Mutlu. 2016. Anticipatory robot control for efficient human-robot collaboration. In *2016 11th ACM/IEEE international conference on human-robot interaction (HRI)*. IEEE, 83–90.
- [26] Kevin Huang, Jiannan Li, Mauricio Sousa, and Tovi Grossman. 2022. ImmersivePOV: Filming How-To Videos with a Head-Mounted 360° Action Camera. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) (CHI '22). Association for Computing Machinery, New York, NY, USA, Article 91, 13 pages. <https://doi.org/10.1145/3491102.3517468>
- [27] Niels Joubert, Dan B Goldman, Floraine Berthouzoz, Mike Roberts, James A Landay, Pat Hanrahan, et al. 2016. Towards a drone cinematographer: Guiding quadrotor cameras using visual composition principles. *arXiv preprint arXiv:1610.01691* (2016).
- [28] Wendy Ju. 2015. The design of implicit interactions. *Synthesis Lectures on Human-Centered Informatics* 8, 2 (2015), 1–93.
- [29] Juho Kim, Phu Tran Nguyen, Sarah Weir, Philip J. Guo, Robert C. Miller, and Krzysztof Z. Gajos. 2014. Crowdsourcing Step-by-Step Information Extraction to Enhance Existing How-to Videos. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) (CHI '14). Association for Computing Machinery, New York, NY, USA, 4017–4026. <https://doi.org/10.1145/2556288.2556986>
- [30] Ryan Knott. 2022. Video statistics, habits, and trends you need to know. <https://www.techsmith.com/blog/video-statistics/>
- [31] Jiannan Li, Mauricio Sousa, Chu Li, Jessie Liu, Yan Chen, Ravin Balakrishnan, and Tovi Grossman. 2022. ASTEROIDS: Exploring Swarms of Mini-Telepresence Robots for Physical Skill Demonstration. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) (CHI '22). Association for Computing Machinery, New York, NY, USA, Article 111, 14 pages. <https://doi.org/10.1145/3491102.3501927>
- [32] Jian Liao, Adnan Karim, Shivesh Jadon, Rubaiat Habib Kazi, and Ryo Suzuki. 2022. RealityTalk: Real-Time Speech-Driven Augmented Presentation for AR Live Storytelling. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology (UIST '22)*. Association for Computing Machinery, New York, NY, USA.
- [33] Christian Licoppe. 2017. Skype appearances, multiple greetings and 'coucou': The sequential organization of video-mediated conversation openings. *Pragmatics* 27, 3 (2017), 351–386.
- [34] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Ubowaja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, et al. 2019. Mediapipe: A framework for building perception pipelines. *arXiv preprint arXiv:1906.08172* (2019).
- [35] Lucy Macken and Paul Ginns. 2014. Pointing and tracing gestures may enhance anatomy and physiology learning. *Medical teacher* 36, 7 (2014), 596–601.
- [36] Cynthia Matuszek, Liefeng Bo, Luke Zettlemoyer, and Dieter Fox. 2014. Learning from unscripted deictic gesture and language for human-robot interactions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 28.
- [37] Richard E Mayer, Logan Fiorella, and Andrew Stull. 2020. Five ways to increase the effectiveness of instructional video. *Educational Technology Research and Development* 68, 3 (2020), 837–852.
- [38] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. 2019. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2630–2640.
- [39] Terrance Mok and Lora Oehlberg. 2017. Critiquing physical prototypes for a remote audience. In *Proceedings of the 2017 Conference on Designing Interactive Systems*. 1295–1307.
- [40] Benjamin A Newman, Abhijit Biswas, Sarthak Ahuja, Siddharth Girdhar, Kris K Kitani, and Henny Admoni. 2020. Examining the effects of anticipatory robot assistance on human decision making. In *International Conference on Social Robotics*. Springer, 590–603.
- [41] Tobias Nägeli, Javier Alonso-Mora, Alexander Domahidi, Daniela Rus, and Otmár Hilliges. 2017. Real-Time Motion Planning for Aerial Videography With Dynamic Obstacle Avoidance and Viewpoint Optimization. *IEEE Robotics and Automation Letters* 2, 3 (2017), 1696–1703. <https://doi.org/10.1109/LRA.2017.2665693>
- [42] Nadya Peek, Jennifer Jacobs, Wendy Ju, Neil Gershenfeld, and Tom Igoe. 2021. Making it a distance: teaching hands-on courses during the pandemic. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–5.
- [43] Ken Perlin, Zhenyi He, and Karl Rosenberg. 2018. Chalktalk: A Visualization and Communication Language—As a Tool in the Domain of Computer Science Education. *arXiv preprint arXiv:1809.07166* (2018).
- [44] Daniel Rakita, Bilge Mutlu, and Michael Gleicher. 2018. An Autonomous Dynamic Camera Method for Effective Remote Teleoperation. In *2018 13th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. 325–333.
- [45] Daniel Rakita, Bilge Mutlu, and Michael Gleicher. 2018. RelaxedIK: Real-time Synthesis of Accurate and Feasible Robot Arm Motion. In *Robotics: Science and Systems*. Pittsburgh, PA, 26–30.
- [46] Mose Sakashita, E. Andy Ricci, Jatin Arora, and François Guimbretière. 2022. RemoteCoDe: Robotic Embodiment for Enhancing Peripheral Awareness in Remote Collaboration Tasks. *Proc. ACM Hum.-Comput. Interact.* 6, CSCW1, Article 63 (apr 2022), 22 pages. <https://doi.org/10.1145/3512910>
- [47] Kenji Sakita, Koichi Ogawara, Shinji Murakami, Kentaro Kawamura, and Katsushi Ikeuchi. 2004. Flexible cooperation between human and robot by interpreting human intention from gaze information. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*(IEEE Cat. No. 04CH37566), Vol. 1. IEEE, 846–851.
- [48] Nazmus Saquib, Rubaiat Habib Kazi, Li-Yi Wei, and Wilmot Li. 2019. Interactive Body-Driven Graphics for Augmented Video Performance. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) (CHI '19). Association for Computing Machinery, New York, NY, USA, 1–12. <https://doi.org/10.1145/3290605.3300852>
- [49] Allison Sauppé and Bilge Mutlu. 2014. Robot deictics: How gesture and context shape referential communication. In *2014 9th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 342–349.
- [50] Emmanuel Senft, Michael Hagenow, Pragathi Praveena, Robert Radwin, Michael Zinn, Michael Gleicher, and Bilge Mutlu. 2022. A Method For Automated Drone Viewpoints to Support Remote Robot Manipulation. *arXiv preprint arXiv:2208.04391* (2022).
- [51] Emmanuel Senft, Michael Hagenow, Robert Radwin, Michael Zinn, Michael Gleicher, and Bilge Mutlu. 2021. Situated Live Programming for Human-Robot Collaboration. In *The 34th Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) (UIST '21). Association for Computing Machinery, New York, NY, USA, 613–625. <https://doi.org/10.1145/3472749.3474773>
- [52] Thomas Sheridan. 1986. Human supervisory control of robot systems. In *Proceedings. 1986 IEEE International Conference on Robotics and Automation*, Vol. 3. IEEE, 808–812.
- [53] Tim J Smith. 2012. The attentional theory of cinematic continuity. *Projections* 6, 1 (2012), 1–27.
- [54] Mauricio Sousa, Daniel Mendes, Rafael Kuffner Dos Anjos, Daniel Medeiros, Alfredo Ferreira, Alberto Raposo, João Madeiras Pereira, and Joaquim Jorge. 2017. Creepy Tracker Toolkit for Context-aware Interfaces. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces* (Brighton, United Kingdom) (ISS '17). ACM, New York, NY, USA, 191–200. <https://doi.org/10.1145/3132272.3134113>
- [55] Ryo Suzuki, Rubaiat Habib Kazi, Li-yi Wei, Stephen DiVerdi, Wilmot Li, and Daniel Leithinger. 2020. RealitySketch: Embedding Responsive Graphics and Visualizations in AR through Dynamic Sketching. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology* (Virtual

- Event, USA) (*UIST '20*). Association for Computing Machinery, New York, NY, USA, 166–181. <https://doi.org/10.1145/3379337.3415892>
- [56] Stefanie Tellex, Nakul Gopalan, Hadas Kress-Gazit, and Cynthia Matuszek. 2020. Robots that use language. *Annual Review of Control, Robotics, and Autonomous Systems* 3, 1 (2020).
- [57] Katherine M Tsui, Munjal Desai, Holly A Yanco, and Chris Uhlik. 2011. Exploring use cases for telepresence robots. In *2011 6th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 11–18.
- [58] Sylvaine Tuncer, Barry Brown, and Oskar Lindwall. 2020. On Pause: How Online Instructional Videos Are Used to Achieve Practical Tasks. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) (*CHI '20*). Association for Computing Machinery, New York, NY, USA, 1–12. <https://doi.org/10.1145/3313831.3376759>
- [59] Ana M Villanueva, Ziyi Liu, Zhengzhe Zhu, Xin Du, Joey Huang, Kylie A Pepler, and Karthik Ramani. 2021. RobotAR: An Augmented Reality Compatible Teleconsulting Robotics Toolkit for Augmented Makerspace Experiences. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) (*CHI '21*). Association for Computing Machinery, New York, NY, USA, Article 477, 13 pages. <https://doi.org/10.1145/3411764.3445726>
- [60] Kenta Yamamoto, Yuki Koyama, and Yoichi Ochiai. 2022. Photographic Lighting Design with Photographer-in-the-Loop Bayesian Optimization. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology* (Bend, OR, USA) (*UIST '22*). Association for Computing Machinery, New York, NY, USA, Article 92, 11 pages. <https://doi.org/10.1145/3526113.3545690>
- [61] Yaxi Zhao, Razan Jaber, Donald McMillan, and Cosmin Munteanu. 2022. “Rewind to the Jiggling Meat Part”: Understanding Voice Control of Instructional Videos in Everyday Tasks. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) (*CHI '22*). Association for Computing Machinery, New York, NY, USA, Article 58, 11 pages. <https://doi.org/10.1145/3491102.3502036>

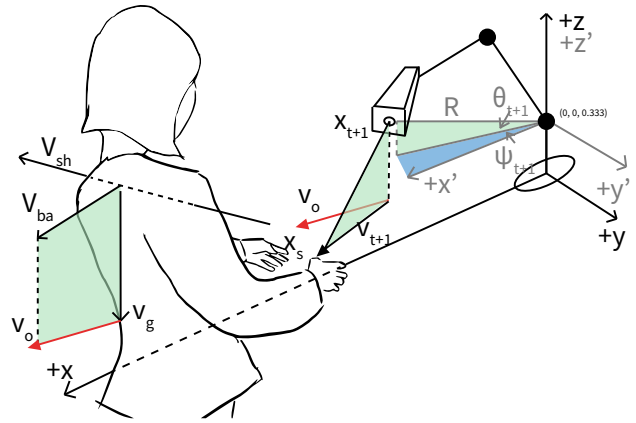


Figure 16: The computation of the desired camera direction v_o (in red) and the polar coordinate system (axes in grey) we used for optimization. Angles in vertical planes are in green, angles in horizontal planes are in blue.

system is parallel to the robot coordinate system. R is the distance between the camera and the origin. θ is the angle that is on the vertical plane. ψ is the angle that is one the horizontal plane. We chose $R \in (0.36m, 0.66m)$, $\theta \in (-\pi/10, 0.4\pi)$, and $\psi \in (-0.4\pi, 0.4\pi)$ as the bounds in optimization.

A ADDITIONAL DETAILS ON ROBOT MOTION PLANNING

We provide more details on motion planning, in addition to the content in 5.4, including the calculation of the orientation cost, and the coordinate system we used in optimization.

A.1 Orientation Cost

As introduced in 5.4.1, for the orientation cost term we calculate it based on the different between the orientation of the next camera position, v_{t+1} and the target camera orientation v_o . Both vectors are in the ground plane (x - y plane in Figure 16). We describe the procedure to compute these two vectors for the case that the camera should look perpendicular at the link between the instructor’s two shoulder joints (v_{sh} indicates its direction). The computation for truck movements, where the camera orientation should be perpendicular to the finger trace, is similar.

To compute v_o , we first get one particular vector, $v_{ba} = v_{sh} \times v_g$, that is perpendicular to v_{sh} . $v_g = (0, 0, -1)$ is the gravity direction. Then v_o is the projection of v_{ba} onto the ground plane, normalized. To compute v_{t+1} , we simply project the camera orientation $(x_s - x_{t+1}) / \|x_s - x_{t+1}\|$ onto the ground plane. Then the orientation cost is square of the norm of the difference between these two vectors. That is $c_o = \|v_{t+1} - v_o\|^2$

A.2 Polar Coordinate System Used for Constrained Optimization

To simply the constraint condition in the optimization, we convert the next position of the camera from Cartesian coordinates to polar coordinates. The origin of our polar coordinate system is at the second joint of the robot (Figure 16). The Cartesian coordinates of this point in the robot frame is $(0, 0, 0.333m)$, according to the specification of the Panda robot. The x, y, z axis of the polar coordinate