Robotics: Science and Systems 2019
Freiburg im Breisgau, June 22-26, 2019

Remote Telemanipulation with Adapting Viewpoints in Visually Complex Environments

Daniel Rakita, Bilge Mutlu, Michael Gleicher
Department of Computer Sciences, University of Wisconsin–Madison
{rakita,bilge,gleicher}@cs.wisc.edu

***Abstract*—In this paper, we introduce a novel method to support remote telemanipulation tasks in *complex* environments by providing operators with an enhanced view of the task environment. Our method features a novel viewpoint adjustment algorithm designed to automatically mitigate occlusions caused by workspace geometry, supports visual exploration to provide operators with situation awareness in the remote environment, and mediates context-specific visual challenges by making viewpoint adjustments based on sparse input from the user. Our method builds on the *dynamic camera* telemanipulation viewing paradigm, where a user controls a manipulation robot, and a camera-in-hand robot alongside the manipulation robot servos to provide a sufficient view of the remote environment. We discuss the real-time motion optimization formulation used to arbitrate the various objectives in our shared-control-based method, particularly highlighting how our occlusion avoidance and viewpoint adaptation approaches fit within this framework. We present results from an empirical evaluation of our proposed occlusion avoidance approach as well as a user study that compares our telemanipulation shared-control method against alternative telemanipulation approaches. We discuss the implications of our work for future shared-control research and robotics applications.**

I. Introduction

From an early age, people develop an innate ability to adapt their viewpoints to plan and coordinate manipulations within their environments [40]. People shift how they look at an object throughout a grasping action [34], scan their environments to plan future actions [17], and naturally adjust their viewpoints to look over and around occlusions when handling items in visually cluttered settings [23]. The tight coupling between manipulation and viewpoint contributes to people's adeptness in executing tasks in complex day-to-day environments, such as when crouching to look in a cabinet below a sink to adjust a valve, moving the head to look around in a cluttered cabinet, or looking up to secure a light bulb into a ceiling fixture.

While much work in remote telemanipulation has focused on the control aspects of the problem [27], such as studying effects of time delays [26], impedance [11], and stability [19], little is known about how the operator's viewpoint should *adapt* given environmental and task considerations. In fact, many telemanipulation systems utilize static cameras, where viewpoints are immutable, or use an end-effector camera on the manipulator itself, where the viewpoint and manipulation points are locked together and cannot adapt separately given the task at hand. Recent work has shown the efficacy of moving the camera to continuously adjust the viewpoint on-the-fly for a remote operator [1, 25, 30], leading to telemanipulation performance and perceptual benefits over an array of static cameras and an end-effector camera [30]. However, it still remains unclear *how* the viewpoint should be adapted to afford effective manipulations in *visually complex* environments, *i.e.*, environments where occlusions are likely to occur, where operators may need to look around to obtain situation awareness and plan future actions, or when specific viewpoints may be necessary given the semantics of the task.

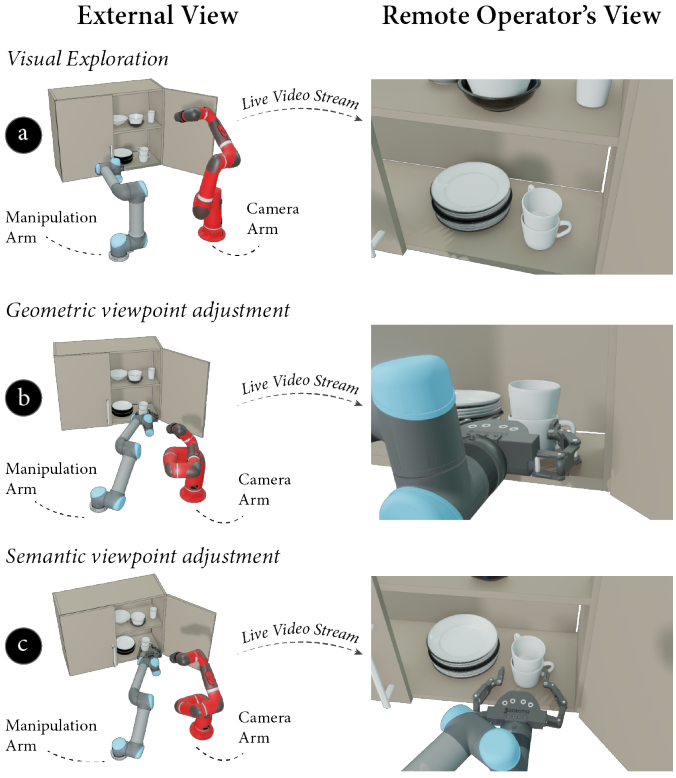

Fig. 1. Building on a *dynamic camera* telemanipulation viewing paradigm, our work supports several viewpoint-related tasks, including (a) visual exploration, (b) geometric viewpoint adjustment, and (c) semantic viewpoint adjustment.

In this paper, we introduce a telemanipulation method where the viewpoint continuously *adapts* over time to better serve manipulations in response to current environment or task conditions. Consider the scenario of a teleoperator remotely preparing a meal for a family member where our method effectively coordinates the operator's manipulations with their viewpoints, such as allowing the user to visually explore the environment to look for cooking oil, automatically providing a sufficient view to reach into a drawer to get a measuring

cup, and accepting viewpoint modifications by the operator to check if the oil has been filled up to the measurement line.

To adapt viewpoints in real-time, our method builds on the *dynamic camera* telemanipulation viewing paradigm, where a user controls a *manipulation robot*, and a *camera robot* servos with a camera-in-hand to provide a view of the manipulation to the operator [30] (Figure I). We adopt the control interface presented in this prior work, where the user fluidly controls the translations and rotations of the manipulation robot's end-effector using a motion controller, and the system automatically adjusts the control frame on-the-fly such that inputs can be made with respect to what is currently seen on screen. In contrast to Rakita et al. [30], our work considers how the two robots should coordinate *together* given environmental and task considerations, both in mathematical formulation by considering the two robots within a single motion optimization and in interface design by simultaneously updating both robots.

To determine *how* the viewpoint should be adapted given environment and task considerations, we draw inspiration from how *people* adjust their viewpoints in complex, human-centered environments. We analyzed a video dataset consisting of people completing tasks in their own kitchens, recorded through head-mounted cameras and identified three distinct viewpoint adaptation behaviors during manipulation (explained in §III). The real-time camera and manipulation-arm control problem is structured as a shared-control method to reduce the user's cognitive load while allowing sparse manual input for user-directed viewpoint shifts, and the shared-control is formulated as a real-time optimization problem to allow fast and real-time coordination between the arms (outlined in §V).

In §VI, we present two evaluations that show the efficacy of our proposed methods. Our first evaluation assesses the performance of our automatic occlusion avoidance method, outlined in §IV, and shows that our occlusion avoidance algorithm is robust in finding effective viewpoints in the presence of visual obfuscations. Our second evaluation features a user study that compares our methods against alternative telemanipulation approaches.

Our contributions in this work include (1) a model of *how* viewpoints should *adapt* in complex environments to support effective telemanipulation, influenced by how *people* adapt their viewpoints in such environments; (2) a set of motion optimization methods and a control interface that support these classes of viewpoint adaptation types; and (3) empirical evaluations that provide insight into the performance and efficacy of the proposed methods.[1]

II. Related Works

Our approach build on ideas from active vision and visual servoing, and is influenced by work in computer graphics.

Active Vision—The control of viewpoint for robotics applications is often termed *active vision*, see Chen et al. [7] or Bajcsy et al. [4] for surveys. Methods reason about posing cameras for numerous applications, such as object search [35], object modeling [5, 8], robot grasp planning [24], object tracking [6], and surveillance [36]. Our work shares similar goals with active-vision surveillance methods, such as maximizing visual coverage and avoiding occlusions [2, 36]. However, this body of work uses mobile-robot platforms and static cameras to survey a wide search area, while our work is focused on viewing a workspace for teleoperation using a camera-in-hand robot. Work in laparoscopic robotic surgery allows surgeons to control a flexible robot camera to obtain a sufficient view of the procedure area [21]. Our work similarly seeks to stream back a sufficient view of a workspace. However, because manually controlling both the camera and manipulation tool may require an expert user, such as a surgeon, we explore automated movement of the camera to support novice users.

Another relevant aspect of active vision considers how to move the viewpoint to gain more information about the scene, termed the "next view" problem. The problem has a long history (see Connolly [12]) for an early example or Zhang et al. [41] for a recent example that considers occlusion. Our work considers how to find improved views for human viewers, rather than views that add information for 3D reconstruction.

Recent work considers how to choose viewpoints for robots to perform tasks. For example, Saran et al. [33] describe methods for determining the most useful viewpoint for performing actions by observing the differences between successes and failures, while Rosman et al. [32] plan sensor locations based on simulations. In contrast, our work chooses viewpoints for human viewers based on real-time information of the scene.

Visual Servoing—Visual servoing is a robot-control paradigm in which a robot moves based on visual feedback (see the work by Corke [13] for a full introduction). In our work, the camera moves based on both what it sees and a geometric understanding of the manipulation robot. Similar to *eye-in-hand* visual servoing systems (e.g., Wilson et al. [39]), the camera robot provides a view using an end-effector-mounted camera.

Animation & Graphics—Computer Graphics and Animation applications consider the problems of automatic camera control, see Christie et al. [10] for a survey. Gleicher and Witkin [18] introduced the idea of adjusting viewpoint position and orientation based on controls in the image plane. Our work uses this idea of mapping visual goals to camera movements. Virtual camera methods have been developed to avoid visual occlusions with objects in the scene [9]. Our visual-occlusion-avoidance method differs from such approaches as these methods have the benefit of a full geometric understanding of the whole environment and a camera that is free to move anywhere in the scene rather than being constrained by the motion of a robot. In data visualization, many works have considered how to choose viewpoints to best enable viewers to see a data set [e.g., 22, 37, 38], but again, these methods require complete geometry. Galvane [16] reviews many approaches that automatically move a camera around in a virtual scene to achieve various goals. Our work draws on this work on automatic camera control, as we dynamically move a camera in our environment to improve the visibility of remote telemanipulation.

[1]Open-source code for the proposed methods are available at https://github.com/uwgraphics/relaxed_ik.

III. Viewpoint Adaptation Types

Our work aims to support effective remote telemanipulation performance, even in visually complex environments, by adapting viewpoints to task and environment considerations, which raises a key question: *how* should the viewpoint be able to adapt when considering the environment or task?

To explore this question, we drew inspiration from how *people* adjust their viewpoints in complex, human-centered environments. We analyzed the Epic Kitchens video dataset [14] consisting of people completing tasks in their own kitchens, recorded through head-mounted cameras. Our goal was to assess the primary ways in which people adjust their viewpoints in day-to-day life to perform effective manipulations and to support these viewpoint adaptation types in telemanipulation.

The video dataset was independently coded by a trained coder through a two pass process. On the first pass, the coder watched the videos and took notes on any patterns that connect viewpoint changes, manipulations, and the environment. The notes were reviewed after viewing, and related concepts were clustered together into higher-level categories. On the second pass, the coder watched the dataset videos again, particularly watching for viewpoint change patterns that further defined or separated the categories referenced in the notes from pass one.

Upon completion of pass two, three central *viewpoint adaptation types* were identified as high level categories that cover most ways that viewpoints change to support manipulations given the environment. These viewpoint adaptation types are:

(1) *Geometrically dictated viewpoint adaptations.* These adaptations describe any viewpoint change that is influenced by the workspace geometry. Examples from the dataset included shifting the viewpoint up to retrieve a spice on the top shelf in the cabinet, shifting the viewpoint to the side to see around a cereal box to grasp a coffee mug on the other side of the table, or looking down into a drawer to grasp a fork.

(2) *Semantically dictated viewpoint adaptations.* These adaptations describe any viewpoint change that is associated with the semantics of a given task. An example was viewing a toaster from above when making toast in order to see the slots to place the bread. This viewpoint selection involved more than just geometric reasoning; while there are many geometrically un-occluded views around the sides of the toaster, these views would be insufficient given the semantics of the particular task.

(3) *Visual explorations*. These adaptations describe any viewpoint changes that involve looking around the environment to plan future actions. Examples from the dataset included looking around to find the lid for a pan and searching the counter-top to find the next ingredient when making dinner.

IV. Technical Overview

In this section, we provide a high-level description for how we support the three visual adaptation types into our remote telemanipulation method. §V provides detailed mathematical treatments of these descriptions, including how these concepts fit within an overall real-time motion optimization framework.

A. Geometrically Dictated Viewpoint Adaptations

As explained in section §III, *geometrically dictated* viewpoint adaptations occur whenever the workspace geometry somehow influences the set of viewpoints that would support an effective manipulation. For example, when reaching into an open drawer, the concave geometry of the drawer limits the set of appropriate viewpoints to those from *above* the drawer, as opposed to around the side or beneath the drawer.

A key technical problem we address in this work is how to determine an effective viewpoint that is robust in the presence of potentially complex environment geometry. Our premise is, given that the camera is able to be moved by the camera-in-hand robot arm, the camera should be able to *recognize* if the visual target point cannot be seen at the moment, such that the camera robot can dynamically *react* and *adapt* to move the camera to a pose where the visual target can be seen again. We construct a differential adjustment algorithm to seek out a new viewpoint at each update that is estimated to get incrementally closer to mitigating such visual occlusions or obfuscations.

Our solution is based on the observation that, regardless of how complex the environment geometry is, we always know of one occlusion-free path: *the free-space path the manipulator took to get to its current configuration*. Thus, if the end-effector cannot be seen, the main strategy of our algorithm is to incrementally servo the camera robot to *align* the camera with the manipulation arm's approach direction until the end-effector can be seen again. We note that there may be *more* occlusion-free paths for a given environment geometry that would elicit clear views of the end-effector. In our current work, we allow the user to manually nudge the camera toward those alternate viewpoints given their understanding of the task and personal preference. In future work, we will explore more real-time mapping, geometric processing, and data-driven approaches to finding such alternate viewpoints automatically.

Using the observation presented above, our viewpoint search algorithm is structured as follows (as illustrated in Figure IV):

(1) Consider the robot's "manipulation vector," *i.e.*, the vector that points forward along the robot's final wrist joint. Suppose there is an upper allowable bound on the angle between the viewpoint vector and the manipulation vector. This can be thought of as the radius r_o of an outer cone emanating out from behind the end-effector where the camera must be placed within. If the end-effector *can* be seen at update t, increase r_o by some increment, capping this value at some maximum.

(2) If the end-effector *cannot* be seen at update t, decrease r_o some increment, placing a minimum value of r_i (an inner cone radius). We place a minimum because the end-effector itself is opaque, so views too aligned with the manipulation vector will elicit views occluded by the end-effector.

(3) If the end-effector *still* cannot be seen when $r_o = r_i$, move the camera closer to the end-effector.

This adjustment process discussed above repeats at each update, either providing more slack on the outer visibility cone radius r_o when the end-effector can be seen, or squeezing the viewpoint angle in and bringing the camera closer to the

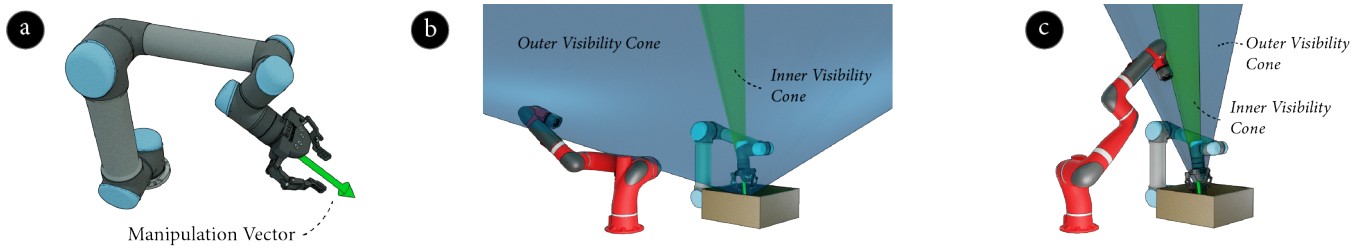

Fig. 2. Illustration of our geometric viewpoint adaptation algorithm. (a) The robot's manipulation vector is used as a proxy for an approach direction. (b) If the end-effector cannot be seen from the camera, (c) the radius of the outer-visibility cone is decreased so that the view aligns more with the manipulation vector.

end-effector until it can be seen. We provide an evaluation of this viewpoint search algorithm in §VI.

B. Semantically Dictated Viewpoint Adaptations

Semantically dictated viewpoint adaptations describe any *context-specific* viewpoint changes. We handle these adaptations by allowing the user to provide sparse manual inputs into the system to specify how they want the viewpoint to be adapted. Specifically, the user can provide a directional input, represented in camera-space, of how they want to adjust the camera position. Because the camera will automatically point at the end-effector visual target, camera-space directional inputs will result in orbital rotations about the visual target point. The directional input can also be made toward or away from the visual target point; thus, these manual modifications can also move the camera in to provide more detail or move the camera out to provide more context given the context of the task.

When the user provides manual directional inputs to the system, we automatically override the geometric search process discussed above and increase the radius r_o. This approach ensures that the user has adequate flexibility when they want to control the viewpoint. When the user stops providing manual inputs, the geometric visual search automatically resumes.

C. Visual Explorations

Visual explorations describe any viewpoint changes that involve looking around and surveying the environment to plan future actions. We handle visual explorations in our telemanipulation method by allowing the user to manually switch to a visual exploration mode, wherein they can naturally adjust the camera's look-at point around the environment. Various automatic aspects of the method are maintained while in visual exploration mode, such as keeping the camera upright and avoiding collisions. When the user decides to exit visual exploration mode, the camera robot smoothly transitions back to automatically looking at the manipulation robot's end-effector.

V. Technical Details

This section provides technical details for the optimization and shared-control solutions outlined above.

A. Motion Optimization Framework

In order to sufficiently realize the manipulation arm and camera arm shared-control method described throughout this work, there are many motion qualities that need to be consistently maintained in the control loop. For example, the manipulation robot should follow end-effector pose goals specified by the user, the camera should look at the visual target point, the manipulation robot and camera robot should not collide, etc.

To accommodate all of these sub-goals in real-time, we use an optimization-based inverse kinematics solver that handles trade-offs between different objectives on the fly. At each update, the method calculates joint angles for each robot that will exhibit these desired features through a process called *inverse kinematics (IK)* (see Aristidou et al. [3] for a review of IK methods. We note that this is a *generalized* IK formulation, because we reason over kinematic goals other than just end effector position and orientation goals.

Our method utilizes the *RelaxedIK* solver to achieve real-time optimization performance [29]. The solver utilizes a flexible non-linear optimization framework to handle IK problems that dynamically trade-off between multiple objectives, and is able to produce per-update motions that accurately follow end-effector pose goals without sacrificing motion feasibility.

The IK problem is formulated as a constrained optimization:

$$\Theta = \underset{\Theta}{\arg\min} \quad \mathbf{f}(\Theta) \text{ s.t. } \mathbf{c}_i(\Theta) \geq \mathbf{0}, \quad \mathbf{c}_e(\Theta) = \mathbf{0} \\ l_i \leq \Theta_i \leq u_i, \forall i \tag{1}$$

Here, Θ is the n–vector of robot joint values (n is the number of degrees of freedom); $\mathbf{c}_i(\Theta)$ is a set of inequality constraints; $\mathbf{c}_e(\Theta)$ is a set of equality constraints; l_i and u_i values define the upper and lower bounds for the robot's joints; and $\mathbf{f}$ is a scalar objective function.

Throughout this work, we consider *both* robot arms together within a single optimization, *i.e.*, the state vector Θ concatenates the joint value degrees of freedom from both arms such that the objectives and constraints can consider both arms together. In prior automatic dynamic camera method by Rakita et al. [30], the manipulation arm and camera arm ran under two separate optimization instances, resulting in behaviors where the camera arm would only react to the actions of the manipulation arm, and the manipulation arm had no sense of the camera arm. Because our current work optimizes over both arms in a single procedure, both arms are aware of each others motion priorities and can plan together. This formulation results in more sophisticated behavior such as the manipulation arm moving its elbow down to clear visual space for the camera.

Our optimization formulation involves twelve objective terms and two constraints. The objective terms encode the following kinematic goals: (1) match end-effector position goal on the

manipulation arm; (2) match end-effector orientation goal on the manipulation arm; (3) minimize joint velocity of the full state vector; (4) minimize joint acceleration of the full state vector; (5) minimize joint jerk of the full state vector; (6) avoid collisions between the arms and modeled environment features; (7) keep camera upright; (8) avoid occlusions caused by the manipulation robot; (9) point camera towards visual target ("look-at" objective); (10) keep camera position between inner and outer visibility cones (outlined in §IV); (11) match desired goal distance between camera and visual target; (12) follow user's manual camera translation inputs (if provided). The two constraints are designed to clamp joint velocities at each update and avoid kinematic singularities for each arm, respectively. Our implementations of objectives 1–8 and both constraints follow prior work [29, 30]; the next section details how we incorporate our viewpoint adaptation types as objectives 9–12.

B. Incorporating Viewpoint Adaptations into RelaxedIK

In this section, we highlight how we incorporate our viewpoint adaptation types, outlined in §IV, into the RelaxedIK optimization framework. We use the same Groove loss function introduced in previous work [29, 30] and specify the loss function parameters for each additional term. Because these terms are incorporated within a larger optimization framework with other objectives built in, other features, such as avoiding occlusions incurred by the manipulation robot, will automatically be exhibited alongside these viewpoint adaptations.

Geometrically Dictated Viewpoint Adaptations. We incorporate geometrically dictated viewpoint adaptations into the RelaxedIK optimization framework using three objective terms, two to encourage the camera position to be between the inner and outer visibility cones outlined in §IV, and one to bring the camera closer if deemed necessary by the search method.

The outer and inner visibility cone terms are:

$$\begin{aligned} \chi_{outer}(\Theta) &= arccos(\ \Lambda(\hat{FK}(\Theta_m)) \cdot \Lambda(\hat{FK}(\Theta_c))\ ) - r_o \\ \chi_{inner}(\Theta) &= r_i - arccos(\ \Lambda(\hat{FK}(\Theta_m)) \cdot \Lambda(\hat{FK}(\Theta_c))\ ) \end{aligned} \tag{2}$$

Here, Θ_m and Θ_c refer to the degrees-of-freedom of Θ corresponding to the manipulation robot and camera robot, respectively; $\hat{FK}(.)$ refers to a function that returns the rotation frame of the end-effector provided a given joint configuration and the forward kinematics model of the arm; and $\Lambda(.)$ refers to a function that returns the "forward" vector of the input rotation frame. These terms use Groove loss parameters of $t = -3.0$, $d = 60.0$, $c = 1e14$, $f = 0.00001$, $g = 10.0$, which encourages both terms to be less than zero.

The camera distance objective is:

$$\chi_{distance}(\Theta) = (\ ||\ FK(\Theta_m)) - FK(\Theta_c))\ ||_2 - d\ )^2 \tag{3}$$

Here, $FK(.)$ is a function that returns the position of the end-effector given a joint configuration and the forward kinematics model of the arm and d refers to a goal distance. This term uses loss function parameters $t = 0.0$, $d = 2.0$, $c = 0.5$, $f = 35.0$, $g = 2.0$, which pulls the objective term output to zero.

Semantically Dictated Viewpoint Adaptations. Semantically dictated viewpoint adaptations are handled in our method by allowing the user to provide a sparse directional input to dictate where the camera should move. As an objective term, this is supported by having the camera pulled toward a new location per update using the following term:

$$\chi_{semantic}(\Theta) = ||\ (\ \mathbf{c} + \lambda \mathbf{g}\ ) - FK(\Theta_c))\ ||_2 \tag{4}$$

Here, $\mathbf{c}$ is the camera location at the previous update and $\mathbf{g}$ is the directional input specified by the user (represented in the camera's local frame). The magnitude of the directional input vector $\mathbf{g}$ and a scalar λ can adjust the sensitivity of the manual inputs. When no manual inputs are being provided from the user, $\mathbf{g}$ is considered to be $[0, 0, 0]^T$. This term uses loss function parameters $t = 0.0$, $d = 2.0$, $c = 0.5$, $f = 35.0$, $g = 2.0$, which pulls the objective term output to zero.

Visual Exploration. Visual exploration is supported in our shared-control method by allowing the user to manually move the visual target and is formulated as the objective term:

$$\begin{aligned} \chi_{lookat}(\Theta) &= dis(\mathbf{t}, \mathbf{v}), \\ \mathbf{v} &= FK(\Theta_c) + \gamma\ \Lambda(\hat{FK}(\Theta_c)) \end{aligned} \tag{5}$$

Here, $\mathbf{t}$ denotes the visual target point, $\mathbf{v}$ denotes the viewpoint vector pointing out of the front of the camera's focal point, $dis(.,.)$ is a function that returns the orthogonal distance between a point and line segment arguments, respectively, and γ is some large scalar value used to cast out the line segment. By default, the visual target point $\mathbf{t}$ is set as the end-effector point on the manipulation robot. However, when users enter visual exploration mode, they are able to move the visual target point $\mathbf{t}$ around by rotating a motion controller. This term uses loss function parameters $t = 0.0$, $d = 2.0$, $c = 0.1$, $f = 10.0$, $g = 2.0$, which pulls the objective term output to zero.

VI. Evaluations

We carried out two forms of evaluation to demonstrate the effectiveness of our dynamic camera shared-control method for remote telemanipulation. Below, we outline our prototype system and discuss the designs and findings of our evaluations.

A. Prototype Details

We instantiated our shared-control camera method in a system, described below, designed to provide sufficient performance and safety to demonstrate its benefits in a user study.

Teleoperation Interface—In our system, we used the *mimicry-control interface*, presented by Rakita et al. [28], to control the manipulation robot for remote teleoperation. This method was shown to be more effective for novice users to control a robot arm using full 6-DOF Cartesian control than other interfaces. We used HTC Vive motion controllers as the motion input devices to capture user input at 80 Hz. One controller moved the manipulation robot while the other allowed for camera translation adjustments using sparse motion controls.

Robots—Our system used a 6-DOF Universal Robots UR5 robot as the manipulation robot and a 7-DOF Rethink Robotics Sawyer robot as the camera robot to match the system used in prior work [30], which served as one of our comparison cases.

TABLE I
ASSESSMENT OF OUR VIEWPOINT ADJUSTMENT ALGORITHM

Scenario	UR5 + Sawyer	Jaco7 Pair
Refrigerator	491/500	500/500
Light bulb	500/500	500/500
Box	494/500	500/500

System Architecture—We set up a distributed system over two computers that utilized the Robot Operating System (ROS) for communication. A Vive motion capture device sent transformation information to the ROS environment through a dedicated Windows computer via UDP messages. A separate computer ran the open-sourced version of the *RelaxedIK* solver,[2] where we incorporated our additional camera objectives and viewpoint adaptation types. In order to achieve sufficient real-time performance to include *both* the manipulation arm and camera arm degrees of freedom within a single optimization structure, we used a version of *RelaxedIK* implemented in the Julia programming language, which is substantially faster than its Python alternative. Solutions are returned in 8 *ms* (125 *Hz*) in our system running on an HP Pavilion laptop with an Intel Core 2.6 *GHz* i7-6700HQ CPU with 32 *GB* RAM. The optimization used the SLSQP solver provided by `NLopt`, and gradients were sent to the solver using forward-mode automatic differentiation using the ForwardDiff Julia package [31].

Our system used two Logitech 930e webcams attached to the camera arm end-effector; one streamed high-definition video over USB that was then displayed on a large-screen monitor, while the other looked for Aruco markers to assess whether the manipulation arm's end-effector could be seen.

B. Assessing Geometrically-dictated Viewpoint Adjustments

In our first evaluation, we designed a testbed of tasks to assess the performance of our proposed viewpoint adjustment algorithm in the presence of occlusions or obfuscations, as outlined in §IV-A. We conducted a procedure in simulation where the camera-in-hand robot would start in a random configuration with the goal of finding a clear view of the manipulation robot's end-effector as quickly as possible, following the procedure outlined in §IV-A.

Tasks—We designed three tasks for our testbed: (1) finding the robot's end-effector as it took something out of a refrigerator on the bottom shelf, requiring a forward view at the level of the bottom of the refrigerator to sufficiently see; (2) finding the robot's end-effector as it screwed a light bulb into a ceiling fixture, requiring a view from below to see adequately; and (3) finding the robot's end-effector as it reached into a box, requiring a view from above to sufficiently see.

Procedure—Our evaluation involved starting the camera robot in 500 random initial configurations for each task. Any initial configuration where the robots did not start in collision states with each other or the environment were deemed acceptable. The manipulation robot started in the same configuration for each of the 500 trials per task, maintaining the same end-effector pose throughout each trial and moving other joints only if redundancy was present and deemed necessary by the optimization as outlined in §V. A trial was deemed successful only if (a) a viewpoint that is not blocked by objects in the environment or by the manipulation arm itself is found in less than ten seconds, determined through ray-casting in the simulated scene; *and* (b) no collisions occurred between the two robots or with statically modeled environment objects.

Robots— We used two separate robot pairs for this evaluation: the UR5 robot as the manipulation arm and the Sawyer robot as the camera robot as well as two 7-DOF Jaco arms.

Results— Our results, summarized in Table I, show that our geometric viewpoint adaptation algorithm found a clear viewpoint on all tasks for both robot pairs in almost all cases. The failures occurred when the random start placed the robots very close to a collision state.

C. User Study

In this section, we present a user study that we compared our telemanipulation shared-control method to other alternatives.

Hypotheses—Our hypotheses predicted that **(H.1.)** our telemanipulation system that considers geometric, semantic, and exploration aspects of viewpoint-manipulation coordination would significantly improve performance and perceptual results over a remote telemanipulation alternative that uses a moving camera without these considerations; and **(H.2.)** a state-of-the-art *co-located* telemanipulation system would outperform our *remote* telemanipulation method on performance and user experience, because participants could utilize depth perception when looking at the environment and more effectively coordinate viewpoint and manipulation by moving their own heads.

Experimental Design—To test our hypotheses, we designed a 3×1 within-participants experiment in which participants completed in-home tasks outlined below using three control paradigms, in a counterbalanced order: *mimicry-control* (MC), *autonomous dynamic camera* (ADC), and viewpoint shared-control manipulation system (VSMS).

(1) Mimicry-control. The user stood behind the robot and guided the robot through motions with their own hand motion [28], and the robot used an optimization-based motion retargeting solution to mimic the user's hand pose motion in real-time. Because the user is co-located with the robot in its workspace, this condition serves as a comparison against how people perform our experimental tasks when they can use their own stereo vision (including depth perception) and control their *own* viewpoints with their heads.

(2) Autonomous dynamic camera. This paradigm followed prior work Rakita et al. [30] that showed the importance of using a moving camera, which outperformed other viewing alternatives for remote manipulations, such as an array of static cameras and an end-effector camera. However, this system did *not* consider environment geometry, handle manual viewpoint shifts given task context, or afford visual explorations.

(3) Viewpoint shared-control telemanipulation system. This paradigm used the methods discussed throughout this work.

Study Setup—To simulate a remote teleoperation setting, the camera robot was placed next to the manipulation robot,

[2]RelaxedIK: https://github.com/uwgraphics/relaxed_ik

and physical dividers separated the participants and robot workspace. Users controlled the robots based on what they saw on the screen. The experimenter sat next to the participants.

Study Tasks—To ensure the generalizability of our findings to a wide range of telemanipulation tasks, we developed three tasks that followed a home-care scenario in which participants would log in to a telemanipulation system to care for a friend or family member by completing the following tasks:

(1) Sock Sorting. Users picked two pairs of white socks from a bin, surrounded by other black socks, and placed them into another bin. This task involved both *geometric* viewpoint reasoning, *i.e.*, maintaining a viewpoint from above with a clear view into the bin, as well *semantic* viewpoint reasoning, *i.e.*, viewing the correct part of the bin to locate white socks.

(2) Table Preparation. Users set the table by retrieving dinner items from a four-cube (2×2) organizer that involved shelves measured $12'' \times 12'' \times 12''$. Participants retrieved a plate from the top left compartment, a fork from the upper right compartment, and a spoon from the lower left compartment. The forks and spoons were placed upright in a cup on their respective shelves. This task also involved *geometric* viewpoint reasoning, *i.e.* maintaining a viewpoint of the end-effector reaching into the compartments, *visual exploration*, *i.e.*, surveying items on shelves, and *semantic* viewpoint reasoning, *i.e.*, refining the viewpoint within the compartments to specify a proper grasp.

(3) Pill Organization. Users picked up a pill bottle and poured a small pill into three containers: a bowl, a cup, and a real pill tray. The containers were chosen to make the task more difficult over time and to show skill level over a single task given a gradient of difficulty. This task involved *semantic* viewpoint reasoning to get a sufficient view of the pouring motion. A variant of this task was also used in prior work [30], allowing us to compare our results to prior work.

Study Procedure—A male experimenter obtained informed consent and provided detail on the study. Participants then viewed a training video on the robot-control approach and the motion controller. Participants then (1) received ten minutes of training on a particular telemanipulation condition using videos and an interactive training session, (2) performed the three tasks outlined in §VI-C using the current condition, and (3) filled out a questionnaire pertaining to the current condition. This process repeated until all conditions were completed, with short breaks between each task as the experimenter reset the robot to its initial configuration and set up the workspace for the new task. Upon completion, participants responded to a demographics survey and received compensation.

Measures—To assess performance, we measured task completion time over the five tasks (*sock sorting*, *table preparation*, and *pill organization* $\times$ 3). For each task, the participants had a maximum time of five minutes. To measure participant perceptions, we administered a questionnaire based on prior research on measuring user preferences and teamwork with a robot [15, 20], including scales on *goal understanding*, *trust*,*ease of use*, *robot intelligence*, *fluency*, and *predictability* (Table II), using a seven-point rating scale.

TABLE II
MEASUREMENTS OF PERCEIVED CONTROL EXPERIENCE

Goal understanding (Cronbach's α = 0.87)
The robot perceives accurately what my goals are
The robot does not understand what I am trying to accomplish
The robot and I are working towards mutually agreed upon goals
Trust (Cronbach's α = 0.86)
I trusted the robot to do the right thing at the right time
The robot was trustworthy
Ease of use (Cronbach's α = 0.93)
The control method made it easy to accomplish the task
I felt confident controlling the robot
I could accurately control the robot
Robot intelligence (Cronbach's α = 0.93)
The robot was intelligent
The robot was able to independently make decisions through the task
The robot had an understanding of the task
The robot had an understanding of my goal during the task
Fluency (Cronbach's α = 0.91)
The robot and I worked fluently together as a team
The robot contributed to the effectiveness of our team
Predictability (Cronbach's α = 0.92)
The robot consistently moved in a way that I expected
The robot's motion was not surprising
The robot responded to my motion inputs in a predictable way

Participants—We recruited 12 participants (5 male, 7 female), aged 19–25 (M = 20.42, SD = 1.67), from a university campus. A post-hoc power analysis with $(1-\beta)$ = .80 and α = .05 found an observed power of 0.96 (d = 3.15) with this sample size. Participants reported low familiarity with robots (M = 2.14, SD = 1.46, measured on a seven-point scale). No participants reported participating in prior robotics research studies. The study took 90 minutes, and each participant received $15 USD.

Results—We analyzed data from all measures using one-way repeated-measures analyses of variance (ANOVA) using control method as the within-participants variable. Figure VI-C shows data and test results from all objective and subjective measures. Our analyses provided full support for both hypotheses.

Discussion—Our results support our hypotheses that our telemanipulation system significantly improves results over the autonomous dynamic camera method on all tasks and many perceptual measures. We observed wide variance in the ADC condition results across all tasks. Because the camera in the ADC condition just moved in response to the motion of the manipulation robot, without any consideration of the task or environment, the resulting motion behavior and resulting viewpoints from the camera could substantially differ across participants, even for the same task. Because our VSMS condition considered the task and environment geometry, the quality of the viewpoint was not dictated by this level of chance, contributing to improved results with lower variance.

We expected mimicry-control to perform better than our method on all tasks and perceptual measures, though we only observe significantly better results on the table preparation task and a marginal effect on the tray pill organization task. We believe that incorporating depth perception into our method will further close the performance gap between the remote and co-located telemanipulation methods on these tasks.

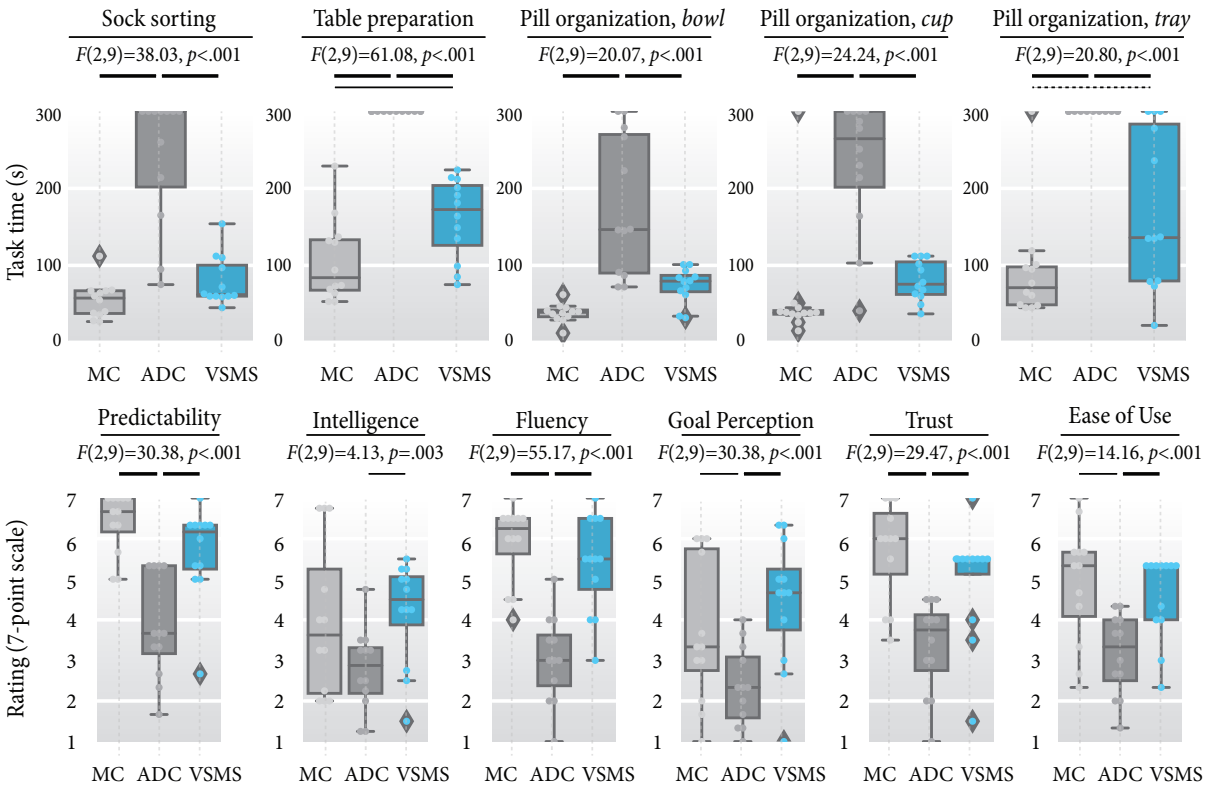

Fig. 3. Results for our user study. Thick, thin, and dotted lines indicate $p < .001$, $p < .05$, and $p < .10$, respectively, in pairwise comparisons.

VII. General Discussion

In this paper, we presented a remote telemanipulation shared-control method where the viewpoint is able to *adapt* to afford effective task execution in complex environments. We introduced a novel viewpoint adjustment algorithm designed to automatically mitigate occlusions caused by the workspace geometry, and showed how we address visual exploration and context-specific visual challenges. In this section, we outline limitations of our current methods, and discuss how our results could be applied to a wide area of robotics applications.

Limitations—Our method has limitations that suggest future extensions. First, our method does not afford depth perception to users using the on-screen interface. We will explore techniques to elicit depth effects, such as using motion parallax or stereo vision, and compare them against mimicry-control where users utilize their own depth perception while manipulating.

Our shared-control camera method benefits the remote user's view without easing the manipulation based on said awareness. We plan to explore ways to use the rich un-occluded data stream to supplement the control algorithm on the manipulation robot. For instance, while our motion optimization framework affords collision avoidance between the two arms and *static* objects modeled ahead of time in the environment, both robots can collide with *dynamic* objects. We will explore ways of providing dynamic collision avoidance given the clear external view of the manipulation point and other parts of the environment.

Our geometric occlusion avoidance algorithm also has known limitations. For instance, the forward "manipulation" vector may not be an accurate proxy for the robot's approach direction in all cases, especially with robots that have a flexible wrist. This limitation could be mitigated by using the *actual* approach direction of the end-effector position over some window of time points. We will investigate such alternatives, as well as explore ways of incorporating mapping, more geometric sensing, and data driven techniques, for finding effective viewpoints.

Conclusion—Our work highlights the potential of using a moving camera that considers the task and environment as part of a robot manipulation system. Our results indicate that an external viewpoint that is able to coordinate with the manipulation point, subject to the environment and task, plays an integral role in manipulation performance. This phenomenon could not only apply to telemanipulation systems but also to fully autonomous systems, where adaptable viewpoints could influence the quality of learned grasp or manipulation policies. We plan to investigate the possible benefits of this viewing paradigm in real-time telemanipulation, shared-control, and supervisory-control settings for applications such as remote home-care, telenursing, or nuclear materials handling, and also explore the methods discussed in this work to inform fully autonomous motion and task policies.

Acknowledgments

The authors would like to thank Olivia Hughes for doing the data coding described in Section III. This research was supported by the National Science Foundation under award 1830242 and the University of Wisconsin–Madison Office of the Vice Chancellor for Research and Graduate Education with funding from the Wisconsin Alumni Research Foundation.

References

[1] Firas Abi-Farraj, Nicolò Pedemonte, and Paolo Robuffo Giordano. A visual-based shared control architecture for remote telemanipulation. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4266–4273. IEEE, 2016.

[2] Besma R Abidi, Nash R Aragam, Yi Yao, and Mongi A Abidi. Survey and analysis of multimodal sensor planning and integration for wide area surveillance. *ACM Computing Surveys (CSUR)*, 41(1):7, 2009.

[3] Andreas Aristidou, Joan Lasenby, Yiorgos Chrysanthou, and Ariel Shamir. Inverse kinematics techniques in computer graphics: A survey. In *Computer Graphics Forum*, volume 37, pages 35–58. Wiley Online Library, 2018.

[4] Ruzena Bajcsy, Yiannis Aloimonos, and John K. Tsotsos. Revisiting active perception. *Autonomous Robots*, 42(2):177–196, feb 2018. ISSN 0929-5593. doi: 10.1007/s10514-017-9615-3. URL http://link.springer.com/10.1007/s10514-017-9615-3.

[5] Joseph E Banta, LR Wong, Christophe Dumont, and Mongi A Abidi. A next-best-view system for autonomous 3-d object reconstruction. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 30 (5):589–598, 2000.

[6] Joao P Barreto, Luis Perdigoto, Rui Caseiro, and Helder Araujo. Active stereo tracking of n ¡ 3 targets using line scan cameras. *IEEE Transactions on Robotics*, 26(3): 442–457, 2010.

[7] Shengyong Chen, Youfu Li, and Ngai Ming Kwok. Active vision in robotic systems: A survey of recent developments. *The International Journal of Robotics Research*, 30(11):1343–1377, 2011.

[8] SY Chen and YF Li. Vision sensor planning for 3-d model acquisition. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 35(5):894–904, 2005.

[9] Marc Christie and Jean-Marie Normand. A semantic space partitioning approach to virtual camera composition. In *Computer Graphics Forum*, volume 24, pages 247–256. Wiley Online Library, 2005.

[10] Marc Christie, Patrick Olivier, and Jean-Marie Normand. Camera control in computer graphics. In *Computer Graphics Forum*, volume 27, pages 2197–2218. Wiley Online Library, 2008.

[11] J Edward Colgate. Robust impedance shaping telemanipulation. *IEEE Transactions on robotics and automation*, 9(4):374–384, 1993.

[12] C. Connolly. The determination of next best views. In *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, volume 2, pages 432–435. Institute of Electrical and Electronics Engineers, 1985. doi: 10.1109/ROBOT.1985.1087372. URL http://ieeexplore.ieee.org/document/1087372/.

[13] Peter I Corke. Visual control of robot manipulators-a review. *Visual servoing*, 7:1–31, 1993.

[14] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. *arXiv preprint arXiv:1804.02748*, 2018.

[15] Fred D. Davis. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Q.*, 13(3):319–340, 1989.

[16] Quentin Galvane. *Automatic Cinematography and Editing in Virtual Environments.* PhD thesis, Grenoble Alpes, 2015.

[17] Eleanor J Gibson. Exploratory behavior in the development of perceiving, acting, and the acquiring of knowledge. *Annual review of psychology*, 39(1):1–42, 1988.

[18] Michael Gleicher and Andrew Witkin. Through-the-lens camera control. In *ACM SIGGRAPH Computer Graphics*, volume 26, pages 331–340. ACM, 1992.

[19] Blake Hannaford. Stability and performance tradeoffs in bi-lateral telemanipulation. In *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, pages 1764–1767. IEEE, 1989.

[20] Guy Hoffman. Evaluating fluency in human-robot collaboration. In *International conference on human-robot interaction (HRI), workshop on human robot collaboration*, volume 381, pages 1–8, 2013.

[21] Louis R Kavoussi, Robert G Moore, John B Adams, and Alan W Partin. Comparison of robotic versus human laparoscopic camera control. *The Journal of urology*, 154 (6):2134–2136, 1995.

[22] H. S. Kim, D. Unat, S. B. Baden, and J. P. Schulze. A new approach to interactive viewpoint selection for volume data sets. *Information Visualization*, 12(3-4):240–256, feb 2013. ISSN 1473-8716. doi: 10.1177/1473871612467631. URL http://ivi.sagepub.com/content/12/3-4/240.abstract?etoc.

[23] Jonathan J Marotta and Timothy J Graham. Cluttered environments: Differential effects of obstacle position on grasp and gaze locations. *Canadian Journal of Experimental Psychology/Revue canadienne de psychologie expérimentale*, 70(3):242, 2016.

[24] Yuichi Motai and Akio Kosaka. Hand–eye calibration applied to viewpoint selection for robotic vision. *IEEE Transactions on Industrial Electronics*, 55(10):3731–3741, 2008.

[25] Davide Nicolis, Marco Palumbo, Andrea Maria Zanchettin, and Paolo Rocco. Occlusion-free visual servoing for the shared autonomy teleoperation of dual-arm robots. *IEEE Robotics and Automation Letters*, 3(2):796–803, 2018.

[26] Günter Niemeyer and Jean-Jacques E Slotine. Telemanipulation with time delays. *The International Journal of Robotics Research*, 23(9):873–890, 2004.

[27] Günter Niemeyer, Carsten Preusche, and Gerd Hirzinger. Telerobotics. In *Springer Handbook of Robotics*, pages

741–757. Springer, 2008.

[28] Daniel Rakita, Bilge Mutlu, and Michael Gleicher. A motion retargeting method for effective mimicry-based teleoperation of robot arms. In *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, pages 361–370. ACM, 2017.

[29] Daniel Rakita, Bilge Mutlu, and Michael Gleicher. RelaxedIK: Real-time Synthesis of Accurate and Feasible Robot Arm Motion. In *Proceedings of Robotics: Science and Systems*, Pittsburgh, Pennsylvania, June 2018. doi: 10.15607/RSS.2018.XIV.043.

[30] Daniel Rakita, Bilge Mutlu, and Michael Gleicher. An autonomous dynamic camera method for effective remote teleoperation. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pages 325–333. ACM, 2018.

[31] J. Revels, M. Lubin, and T. Papamarkou. Forward-mode automatic differentiation in julia. *arXiv:1607.07892 [cs.MS]*, 2016. URL https://arxiv.org/abs/1607.07892.

[32] Guy Rosman, Changhyun Choi, Mehmet Dogar, John W. Fisher IIIl, and Daniela Rus. Task-Specific Sensor Planning for Robotic Assembly Tasks. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2932–2939. IEEE, may 2018. ISBN 978-1-5386-3081-5. doi: 10.1109/ICRA.2018.8460194. URL https://ieeexplore.ieee.org/document/8460194/.

[33] Akanksha Saran, Branka Lakic, Srinjoy Majumdar, Juergen Hess, and Scott Niekum. Viewpoint selection for visual failure detection. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5437–5444. IEEE, sep 2017. ISBN 978-1-5386-2682-5. doi: 10.1109/IROS.2017.8206439. URL http://ieeexplore.ieee.org/document/8206439/.

[34] Sheila Schneiberg, Heidi Sveistrup, Bradford McFadyen, Patricia McKinley, and Mindy F Levin. The development of coordination for reach-to-grasp movements in children. *Experimental Brain Research*, 146(2):142–154, 2002.

[35] Ksenia Shubina and John K Tsotsos. Visual search for an object in a 3d environment using a mobile robot. *Computer Vision and Image Understanding*, 114(5):535–547, 2010.

[36] Garimella SVS Sivaram, Mohan S Kankanhalli, and KR Ramakrishnan. Design of multimedia surveillance systems. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 5(3):23, 2009.

[37] Pere-Pau Vázquez, Miquel Feixas, Mateu Sbert, and Wolfgang Heidrich. Viewpoint Selection using Viewpoint Entropy. In *VMV '01 Proceedings of the Vision Modeling and Visualization Conference 2001*, pages 273–280. Aka GmbH, nov 2001. ISBN 3-89838-028-9. URL http://dl.acm.org/citation.cfm?id=647260.718491.

[38] Ivan Viola, Miquel Feixas, Mateu Sbert, and Meister Eduard Gröller. Importance-driven focus of attention. *IEEE transactions on visualization and computer graphics*, 12(5):933–40, jan 2006. ISSN 1077-2626. doi: 10.1109/TVCG.2006.152. URL http://dl.acm.org/citation.cfm?id=1187627.1187810.

[39] William J Wilson, CC Williams Hulls, and Graham S Bell. Relative end-effector control using cartesian position based visual servoing. *IEEE Transactions on Robotics and Automation*, 12(5):684–696, 1996.

[40] Hanako Yoshida and Linda B Smith. What's in view for toddlers? using a head camera to study visual experience. *Infancy*, 13(3):229–248, 2008.

[41] Shihui Zhang, Yuxia Miao, Xin Li, Huan He, Yu Sang, and Xuezhe Du. Determining next best view based on occlusion information in a single depth image of visual object. *International Journal of Advanced Robotic Systems*, 14(1):172988141668567, jan 2017. ISSN 1729-8814. doi: 10.1177/1729881416685672. URL http://journals.sagepub.com/doi/10.1177/1729881416685672.