A Dynamically Weighted Multi-Objective Optimization Approach to Positional Interactions in Remote-Local Augmented/Mixed Reality

Akshith Ullal
Dept. of EECS
Vanderbilt University
Nashville, TN, USA
akshith.ullal@vanderbilt.edu

Cadence Watkins
Dept. of Mechanical Engineering
Vanderbilt University
Nashville, TN, USA
cadence.watkins@vanderbilt.edu

Nilanjan Sarkar
Dept. of Mechanical Engineering
Vanderbilt University
Nashville, TN, USA
nilanjan.sarkar@vanderbilt.edu

Abstract— **With the rise in remote work culture and increased computing capabilities of head-mounted displays (HMDs), more immersive, collaborative experiences are desired in remote-local mixed/augmented (MR/AR) reality. Photorealistic full-body avatar representations of users in remote workspace interactions have shown to have increased social presence, non-verbal behavior, and engagement. However, a direct mapping of the body pose angles from local to the remote workspace will, in most cases, result in positional errors during human-object interaction, caused by the dissimilarity between remote and local workspaces. Hence the interaction must be retargeted, but it should be retargeted in such a way that the original intent of the body pose should be preserved. However, these two objectives sometimes contradict each other. As a result, a multi-objective optimization (MO) problem can be formulated where the primary objective is to minimize positional errors and the secondary objective is to preserve the original interaction body pose. The current state-of-the-art solution uses an evolutionary computation-based inverse kinematic (IK) approach to solve the MO problem where the weights between the objectives must be set by the user based on trial and error, leading to a suboptimal solution. In this paper, we present a new dynamic weight allocation approach to this problem, where a user has the flexibility to set a chosen minimum error tolerance, and the weights will be distributed between the objectives based on a dynamic allocation algorithm. We have tested the adaptability and robustness of this mechanism on motion captured human animations of varying levels of speeds, error tolerances, and redirections. Compared to the static weighting, the dynamic weighted mechanism shows a net (primary + secondary objective) decrease in error ranging from 20.5 % to 34.4% across varying animation speeds and redirections resulting in decreased positional errors and better pose preservation across interactions.**

Keywords— Collaborative mixed reality, multi-objective optimization (MO), logistic function, remote-local workspaces, inverse kinematics.

I. Introduction

Remote work and interactions have gained significant attention, especially due to the ongoing COVID-19 pandemic, with the most common form of remote interactions being 2D audio-video applications such as Zoom, Facetime, etc. These interactions, however, can be more collaborative and engaging when experienced through mixed/augmented (MR/AR) reality using head-mounted displays (HMDs) [1]. Modern HMDs have the capability to track many user features such as eye gaze, the user's position and orientation, and the user's hands, which can then be mapped onto photo-realistic avatars making the interaction more naturalistic [2], [3]. As we transition into the post-pandemic world, many analysts predict that remote work will be a permanent part of the upcoming hybrid work culture [4]–[6]. Post pandemic, remote interactions are also extending to non-work related areas such as entertainment, education and health [7], [8]. Hence there is a need for and opportunity towards creating more immersive full-body interactions in mixed reality. One important aspect of full-body interaction is the body pose of the avatar representing a user. Body pose has been shown to convey increased social presence and also display many personal traits of humans, making it an important component of a person's non-verbal behavior [9]–[13].

Body pose angles can be captured by color-depth cameras (e.g., Kinect) and then mapped onto an avatar representation of a user in the remote workspace. However, a direct one-to-one mapping of the joint angles of the user to the avatar may result in erroneous positional interaction with the remote workspace due to factors such as differences in physical dimensions between the user and their avatar representation, differences in positions of virtual objects between the remote and local environments, and localization and drift errors. Hence the end effectors, which are in most cases the user's hands, need to be redirected to minimize the error between the end effector and the interaction object (goal target). At the same time, the new pose resulting from the redirection should be as similar to the original pose of the user in the local workspace as possible to preserve important nonverbal cues regarding the person's intent (Fig. 1). However, these two objectives increasingly conflict with each other as the difference between the original and goal targets increases between the remote and local workspaces, making it necessary to find a suitable compromise between the two objectives. In this work, we frame this problem as a multi-objective optimization (MO) problem with the primary objective being to minimize the positional error between the end effector and goal target and the secondary objective being to preserve the original body pose while redirecting the effector to the goal target.

Inverse kinematic (IK) approaches have been used for redirecting a kinematic chain to a goal target position. The human body can be modeled as an interaction of several kinematic chains under specific constraints as shown in Fig. 2, where five individual kinematic chains have been centered around the hip joint. Using kinematic chain models allows IK techniques to redirect any joint towards a goal target. Although there have been various IK techniques proposed to obtain this redirection (described in more detail in Section II), a new approach using memetic evolutionary algorithms has shown to be highly effective in finding valid solutions. This approach [14] has also shown to be flexible, as multiple objectives can be added to the optimization fitness function. However, one of the limitations of this method is that the weights used in the fitness function among the different objectives must be manually found by trial and error, which may lead to a suboptimal solution. The solution space for kinematic chain redirections can vary wildly depending on the extent of the error or difference in the remote local workspace

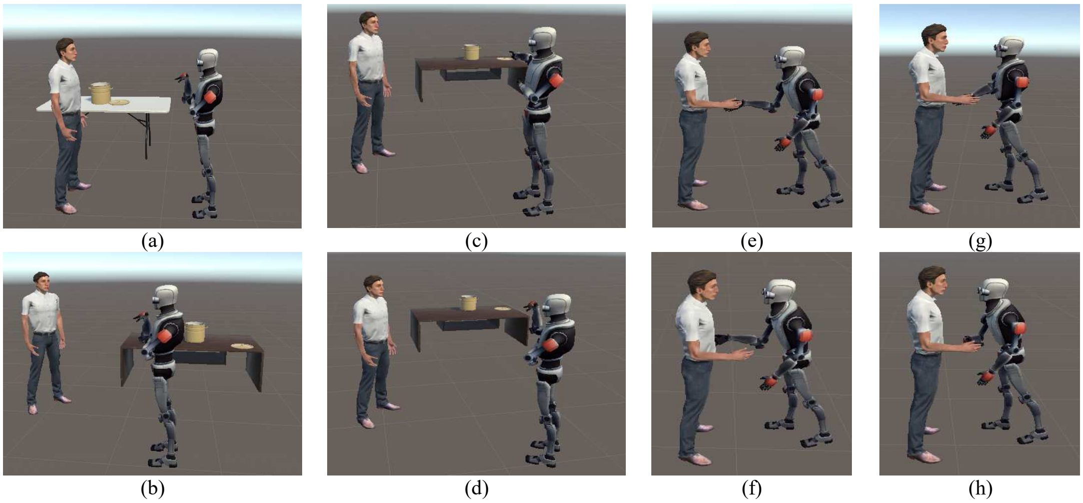


Fig. 1. The motivation for the need of a multi-objective optimization (MO) approach for an interaction involving remote avatars, remote objects (a, b, c, d), and for the specific case of positional interactions(e, f, g, h) in remote-local mixed/augmented reality: (a) Local workspace A - the robot body pose indicates it is talking about the vessel on the table; (b) Remote workspace A - direct body pose mapping from (a) erroneously conveys that the robot is talking about the human due to slight positional changes of human and virtual object from (a); (c) Remote-workspace A – redirection with no optmization belies the orginal intent. Here it seems as if the robot is more authorative when compared to (a) ; (d) Remote workspace A – with a MO approach both the redirection and the body pose non verabal interaction is preserved; (e) Local workspace B – the robot is hesitantly interacting with the human; (f) Remote workspace B – direct body pose mapping results in erroneous interaction, due to slight positional change of the human from (e); (g) Remote workspace B – redirection with no body pose optmization seems the robot is confidently interacting; (h) Remote workspace B – the hesistancy of the robots interaction and its handshake is preserved with a MO based body pose optmization.

environment. For example, redirection due to slight changes in physical size between the user and their avatar will be minor when compared to redirections where the interaction object is placed 180-degree opposite (relative to the user) between the remote and local workspaces. Hence there is a need for dynamically adjusting the weights of each objective so that an improved solution can be obtained.

In this paper, we present a dynamic weight allocation approach that optimizes between the objectives of body joint redirection and preserving the body pose of the original interaction. The presented approach is flexible in that the user can specify a minimum error tolerance, which will then be used for dynamically calculating objective weights. We compare the performance of this dynamic approach with the current state-of-the-art static weight allocation mechanism [14] for different types of configurations, speeds and redirections. The results are shown on motion captured human animation. We test for two general cases of error tolerance: when the margin of error required by the user is 1) strict, and 2) lenient. The main contributions of this paper can be summarized as: i) the development of a dynamically weighted MO approach for positional interactions in remote-local MR/AR environments; ii) the demonstration of the flexibility of the approach according to the user's requirements of minimum tolerance of error, and iii) superior performance of the approach over the current state of the art static approach.

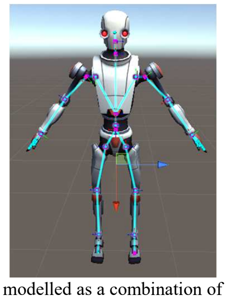


Fig. 2. Body pose modelled as a combination of 5 serial kinematic chains with constraints.

II. Related Work

A. Remote-Local AR/MR Interactions

There are two types of data that can be communicated across remote-local MR/AR systems, namely data related to the workspace and human-oriented data. Systems that can not only send these two types of data but also preserve the interactions between the two are desirable. During its early development, AR involved expert-novice scenarios where a video feed was sent from the remote worksite and the expert provided guidance, mainly through audio and annotations [15]. These types of expert-novice scenarios have been extended to involve multiple users, situated both in the co-located and remote workspace to be able to view the interaction environment [16]. Systems have also been developed that send both a headset's field of view (FOV) and 360-degree panoramic videos captured using mobile devices that allow users to communicate with each other using hand gestures[17]. This is especially useful when collaboration takes place outdoors. There have also been systems where multiple RGBD cameras capture a complete room scene and transport it across to the remote side [18]. In other situations, where human gestures play a more important part than the workspace, mixed reality has been used as a platform for common remote meetings, conducting multi-modal therapy activities[19], [20] and biometric identification [21]. These have included setups where only the head gaze has been redirected [22] to collaborative sketching using avatar bust representations [1], [23] where users' position, orientation, and hands have been tracked [24], [25]. The avatars have also been customized according to the user's nonverbal behaviors to increase their sense of social presence [26]. There are two common ways to convey full-body non-verbal behavior. First, the RGBD data of the human can be captured using a 3D point cloud, which are then transported and replayed on a remote device. Second, human data can be overlayed onto a 3D model to create photo-realistic avatars. Although the 3D point cloud is realistic and accurately depicts the person, these data cannot be manipulated or redirected to the correct locations in the remote environments. Hence error correction using a photorealistic avatar is preferred. In terms of preserving the workspace and gestures, recent work includes "Mini-me" [27], where gaze, body orientation and hand redirection are explored in mixed reality. However, the redirection in [27], only considers that of the hand (as a 2 joint manipulator) and not that of the whole body. Moreover, it is not generalizable to any joint on the body. The generalized framework for full-body interactions has been provided in [14] (which our paper is based on), where the whole body is modeled using kinematic chains and its redirection is considered to multiple goal targets.

B. *Inverse Kinematics and Optimization*

Humanoid characters are extensively used in computer games and animations, where they perform a variety of common motions such as walking and running. Many of these animations have been modeled from motion capture systems. A very common application in these games and animations is the inverse kinematics problem, i.e., given a goal target, what joint configurations of the humanoid will make the end effector reach that goal target. Obtaining a solution to the inverse kinematics problem for a humanoid modeled using kinematic chains is not straightforward, and a unique solution is not guaranteed to exist. The complexity of the problem also increases with the increase in the degrees of freedom (DOF) of the kinematic structure. On top of this, naturalistic human motion has many joint and soft motion constraints making the IK problem a highly nonconvex one. Several approaches have been proposed to solve IK problems that include analytic, heuristic, gradient-based, sampling-based, and learning approaches, where each method performs well for a specific type of application [14]. For a given kinematic geometry, analytic methods provide solutions in closed-loop form that are exact and return all the solutions for any joint configuration. Although exact solutions may be available using analytic methods, they are only practical for simple geometries due to the increase in complexity for each additional DOF. For complex kinematic geometries, the focus has been on heuristic and iterative methods. These methods are fast and can be used for real-time applications. Well known heuristic algorithms include cyclic co-ordinate descent (CCD) [28] and Forward and Backward Reaching Inverse Kinematics (FABRIK) [29]. These heuristic algorithms are also implemented in modern game engines such as Unity, Unreal and Maya. One of the drawbacks of using heuristic solutions is that they only optimize for position and are not efficient if the goal is for a kinematic chain to reach a particular orientation. This problem is overcome by gradient-based IK, which involve the computationally expensive calculation of first or second order derivatives and operate directly in joint space. Recently, evolutionary computing-based IK solutions have found to be effective. In [14], a hybrid implementation of genetic algorithm (GA) and particle swarm optimization (PSO) have been compared with analytic, heuristic and gradient-based methods and shown to be highly effective in finding valid solutions. It has also been shown to be flexible in satisfying position, orientation, and displacement objectives while finding the solutions.

III. PROBLEM FORMULATION AND OPTIMIZATION FRAMEWORK

In order to represent the human body's pose, we have used the model as shown in Fig. 2, which consists of 5 serial kinematic chains with constraints and 20 joints [14]. A body pose can be represented as a joint variable configuration θ

$$\theta = (j_1 | j_2 | j_3 | \dots\dots\dots | j_{n-1} | j_n) \tag{1}$$

where n represents the number of joints in the model and $j_1, j_2 \; \dots . j_n$ represent the individual 3D joint angles. Each of the joints also have realistic human body pose constraints

$$\theta_{i_{min}} \leq j_i \leq \theta_{i_{max}} \quad \forall i = 1, \dots\dots . n \tag{2}$$

where θ_{imin} and θ_{imax} are the minimum and maximum angles for the i_{th} joint. The goal of our optimization framework is to find θs for the remote workspace such that the primary objective error between the end effector and goal target is minimized, i.e., it should be within the minimum error tolerance specified by the user. Once this primary objective is satisfied, among these θs, the one set of angles that keeps the pose as similar to the local workspace pose, is selected (secondary objective). This MO is carried out for every frame. The primary and secondary objective errors are defined below.

A. *Primary and secondary objective errors*

Several distance metrics have been proposed to compare body poses [30]. Among these, we choose the weighted joint co-ordinate distance (WJCD) metric to compare different body pose interactions since it is based on Euclidean distances, the same metric that we use to measure our input errors for positional interactions. This is important because in MO problems, if the objectives have differing units and scale, these values have to be normalized to a common unit and scale, which is not a trivial task [31]. For example, if one objective is measured in Euclidean distance and the other in radian angles, the distance error not being bounded, tends to overshadow the angle error which is bounded between 0 and 2π , causing a difficult normalization problem. Since positional errors are already calculated in Euclidean distances, using WJCD we can quantify body poses (secondary error) also using Euclidean distances, preventing the normalization problem. The primary objective error (x_{pe}) can be defined as

$$x_{pe} = \alpha \sum_{i=1}^{n_p} d_E^2 \,(p_i, g_i) \tag{3}$$

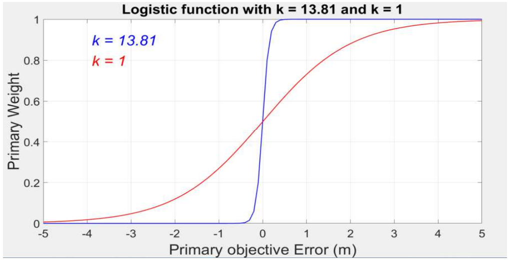


Fig. 3. Variations of the function growth rate value (*k*) in the logistic function. All other variables between (*k* = 13.81) and (*k* = 1) are the same.

where $p_i, g_i \in R^3$ represent the Cartesian co-ordinates of the primary objective for joint *i* of the current candidate solution of the optimization framework and its respective goal target, respectively. $d_E\ (p_i\ , g_i)$ represents the Euclidean distance between p_i , g_i and n_p represents the number of primary objectives. The primary error is multiplied by the primary weight (calculation described in a later section). Similarly, the secondary objective error (x_{se}) based on WJCD can be defined as

$$x_{se} = \beta \sum_{i=1}^{n_s} d_E^2\ (p_i^l\ , p_i^r) \tag{4}$$

where p_i^l, $p_i^r \in R^3$ represent the Cartesian co-ordinates of the secondary objective for joint i in local and remote workspace, respectively. $d_E\ \left(p_i^l\ , p_i^r\right)$represents the Euclidean distance between p_i^l, p_i^r and n_s represents the number of secondary objectives. The secondary error is multiplied by the secondary weight β (calculation described in a later section). To obtain p_i^r, we first record the temporary joint positions $(p_1^t\ , p_2^t \dots \dots p_{n-1}^t\ , p_n^t)$ given by the optimization framework that minimizes the primary error in the remote workspace (5)

$$f(\theta_{local}) = (p_1^t\ , p_2^t \dots \dots p_{n-1}^t\ , p_n^t) \tag{5}$$

If p_0^l is the root node (in this case, the hip joint) position in the local workspace co-ordinates and p_{root}^r is the root node in remote workspace co-ordinates, then we can get the offset position p_{off} between the two workspace avatar root nodes as:

$$p_{off}\ = p_{root}^r - \ p_0^l \tag{6}$$

we can then compute p_i^r as follows:

$$p_i^r\ = \ p_i^t + (p_{off_x}, p_{root_y}^r, p_{off_z}) \quad \forall i = 1, \dots \dots . n \tag{7}$$

where p_{off_x}, p_{off_z} are the x and z co-ordinates of the remote-local workspace offset, and $p_{root_y}^r$ is the y-coordinate (height) of the root node of the remote workspace avatar.

B. *Cost function and Optimization framework*

We choose the cost function to obtain smooth optimization solution that does not exhibit sudden fluctuations for small changes in objective errors. To this end, we have used the method of combining the multiple objectives into a single scalar root mean squared (RMS) value shown in (8,9). By representing the cost function ($C(x)$) as an RMS equation, we convert the optimization problem from a linear to a quadratic one, which has an approximate convex solution.

$$C(x) = \sqrt{\frac{\alpha\ \sum_{i=1}^{n_p} d_E^2\ (p_i, g_i) + \beta \sum_{i=1}^{n_s} d_E^2\ (p_i^l\ , p_i^r)}{n_p + n_s}} \tag{8}$$

which can be simplified and written as

$$C(x) = \sqrt{\frac{x_{pe} +\ x_{se}}{n_p + n_s}} \tag{9}$$

where x_{pe} and x_{se} represent primary and secondary errors, respectively.

IV. Dynamic Weighting Function

The GA-based IK architecture used in [14] has a manual weighting scheme, i.e., the weights have to be adjusted according to the user preference. This is mainly done by trial and error since finding the exact relation between the objective weights is complex and time-consuming. We present a new dynamic weight allocation approach to this problem, where the weights are distributed dynamically between the primary and secondary objectives based on a user-specified tolerance. In this section, we describe the function used for the weight allocation and the factors that affect it.

A. *Logistic function*

We chose to use the standard logistic function, to dynamically allocate weights between the objectives, which is shown in (10):

$$f(x) = \frac{L}{1 + e^{-k(x-x_0)}} \tag{10}$$

where L represents the maximum function value, *k* represents the function growth rate and $x_0$represents the *x* value at the midpoint of *f(x)*. In our application, *f(x)* represents the weight for the primary objective, *x* represents the primary objective error, and *k* is used to control the steepness of the curve. Although there are many dynamic easing functions that can be used, the logistic function is the most widely used due to its smoothness and monotonic nature, which smoothly transitions the weights between the objectives [32]. Logistic functions have been used in many optimization and machine learning applications [33].

The two main factors that influence the shape of the logistic function are x_0 and *k*. *x* is the primary error and is considered as the input to the logistic function. Hence the domain of the function is $(0, \infty)$ and the default range is [0.5*L*, 1*L*], when $x_0 = 0$. As x_0 is increased the *x* value of *f(x)* at the midpoint increases. The value of *k* is used to control the strictness of the allowed error. Fig. 3 shows the effect of different *k* values keeping all other variables the same. When *k* = 13.81, full weightage is given to the primary objective after *x* ≈ 0.2m whereas when *k* = 1.00, full weightage is given to the primary objective after *x* ≈ 5m. As for *L*, the absolute value of the weight does not matter, but the relative distribution of weights among the objectives counts.

V. PRIMARY AND SECONDARY WEIGHTS

All joints considered as part of the primary objective will be multiplied by the primary weight and similarly, all joints considered part of the secondary objective will be multiplied by the secondary weight in the cost function (8,9). The primary weight is set based on the average error of the primary objective joints x_{pavg} and the user-defined minimum error tolerance k_{tol} (described in the next section).

$$x_{pavg} = \frac{\sum_{i=0}^{n_p} d_E(p_{if}, g_{if})}{n_p} \tag{11}$$

where $p_{if}, g_{if} \in R^3$ represent the Cartesian coordinates of the primary objective for joint i and its respective goal target. $d_E\left(p_{if}, g_{if}\right)$ represents the Euclidean distance between p_i , g_i, and n_p represents the number of primary objectives. It is important to note the difference between x_{pavg} in (11) and x_{pe} in (3). x_{pavg} is calculated once every frame and is used to set the primary weight that is then used to calculate x_{pe} , which is the error for the current candidate solution of the optimization framework cost function, calculated many times in a frame and used to find the best joint angle configuration. The primary weight (α) is obtained by from equation (12). Once the primary weight is determined we can calculate the secondary weight (β) as shown in (13)

$$\alpha\ (primary\ weight) = \frac{L}{1 + e^{-k_{tol}(x_{pavg})}} \tag{12}$$

$$\beta\ (secondary\ weight) = L - \alpha \tag{13}$$

VI. USER-DEFINED MINIMUM ERROR TOLERANCE CALCULATION

Different types of avatar-object interactions require different levels of accuracy. For example, if an avatar-object interaction involves lifting a large virtual ball, the error tolerance required may not be very precise when compared to a dexterous manipulation of holding and rotating a cup with one's fingers. The differences in error tolerance required for different tasks and contexts can be set up using our weighting approach. Setting a minimum error tolerance (x_{tol}) amounts to letting the dynamic weighting function know how to decide on the threshold for allocating the maximum weightage to the primary objective. For example, suppose if a highly dexterous application requires a minimum error tolerance of 0.01 m, then, we need to find a k_{tol} value in (12), for which when x_{pavg} > 0.01m the primary weight $(\alpha) \approx L$. This can be done by rearranging (12) as follows:

$$k_{tol} = \frac{\log(\frac{f(x)_{max}}{L(1 - f(x)_{max})})}{x_{tol}} \tag{14}$$

where x_{tol} is the user-defined minimum error tolerance and $f(x)_{max} \approx 1$ to prevent k_{tol} from being indeterminate if we consider $f(x)_{max}$ exactly equal to 1.

VII. METHOD

The rationale we follow to compare the two weighting strategies is to keep all other variables in the testing conditions equal and closely controlled so that the cases only differed in the weighting strategies. Testing for the static weighting case is not straightforward, as a body pose action sequence needs to be tested with all combinations of primary and secondary weights. Hence, we have chosen to test both these weighting approaches using motion captured animation cycles of human actions. In this way, we can have a combination of weights during one animation cycle and then change it for the next cycle knowing that the sequence of body joint movements is exactly the same as the previous animation cycle. In our case, for a positional target location we initially start with the first animation cycle having primary weight (α)= 1L (maximum weight) and secondary weight (β) = 0. Then for every consequent animation cycle we decrement the primary weight by 0.01 until it reaches 0 and correspondingly at the same time secondary weight reaches 1L. We run the dynamic weighting case for the same animation and number of cycles as that of the static weighting case, but with the primary weight now selected according to (12). All other variables including the optimization architecture and avatar models used are the same for both the static and dynamic cases. Subsequently, to evaluate the robustness and adaptability of the dynamic

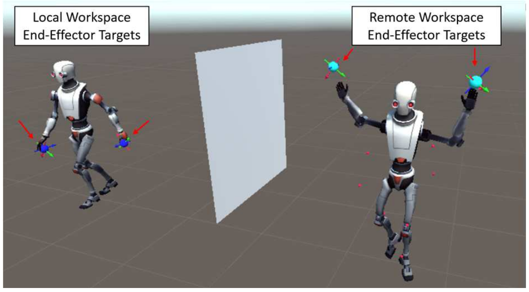


Fig. 4. Simulation setup. The local workspace with positional goal targets (left) and the remote workspace with its positional goal targets(right). The red dots on the remote workspace side are the secondary objective goal targets for the respective joints, obtained from the local workspace pose.

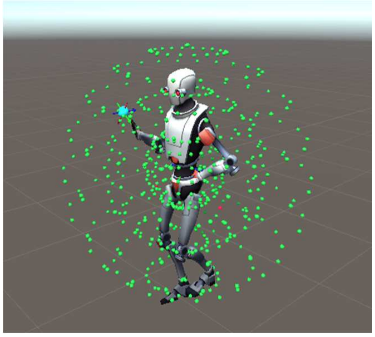


Fig. 5. All positions (represented by green dots) tested for redirection in the remote workspace

TABLE 1
COMPARISON OF THE PRIMARY AND SECONDARY ERRORS (PE and SE) FOR STATIC VS DYNAMIC APPROACH FOR STRICT MINIMUM ERROR TOLERANCE (x_{tol}= 0.25m, $k = 13.81$) CONDITION

Motion capture animation	Average static PE (m)	Average static SE (m)	Average dynamic PE (m)	Average dynamic SE (m)	Reduction in PE – static vs dynamic (%)	Reduction in SE – static vs dynamic (%)	Net (PE + SE) change in error (%)
Idling (1 target)	0.19	0.31	0.05	0.52	71.67	-68.64	3.03
Idling (2 target)	0.39	0.51	0.07	0.67	80.07	-33.10	46.97
Walking (1 target)	0.27	0.58	0.08	0.88	70.67	-53.00	17.67
Walking (2 target)	0.38	0.73	0.11	1.10	70.47	-51.06	19.40
Running (1 target)	0.20	0.74	0.05	0.94	73.02	-27.39	45.63
Running (2 target)	0.41	1.06	0.11	1.37	72.11	-28.33	43.78
Sprinting (1 target)	0.19	0.84	0.05	1.03	73.50	-21.67	51.83
Sprinting (2 target)	0.39	1.18	0.11	1.48	72.25	-25.17	47.08

TABLE II
COMPARISON OF THE PRIMARY AND SECONDARY ERRORS (PE and SE) FOR STATIC VS DYNAMIC APPROACH FOR LENIENT MINIMUM ERROR TOLERANCE (x_{tol}= 5m, $k = 1$) CONDITION

Motion capture animation	Average static PE (m)	Average static SE (m)	Average dynamic PE (m)	Average dynamic SE (m)	Reduction in PE – static vs dynamic (%)	Reduction in SE – static vs dynamic (%)	Net (PE + SE) change in error (%)
Idling (1 target)	0.19	0.30	0.15	0.30	19.57	-0.18	19.38
Idling (2 target)	0.39	0.50	0.32	0.50	17.18	-0.02	17.15
Walking (1 target)	0.19	0.48	0.15	0.47	19.50	3.36	22.86
Walking (2 target)	0.39	0.73	0.32	0.70	17.83	3.24	21.08
Running (1 target)	0.18	0.73	0.15	0.72	14.71	2.26	16.98
Running (2 target)	0.41	1.05	0.32	1.00	22.61	5.13	27.74
Sprinting (1 target)	0.19	0.73	0.14	0.84	27.51	-14.94	12.56
Sprinting (2 target)	0.40	1.19	0.29	1.18	25.68	0.57	26.26

weighting mechanism we have also tested it by varying other factors described in the following sections:

A. Testing with different minimum error tolerances

Ideally, minimum error in the positional interactions (primary objective) is desired during remote-local interactions in MR/AR. The dynamic weighting mechanism is focused on this task i.e., minimizing the primary objective error for the specified minimum error tolerance. Once this is achieved, then it optimizes the secondary objective. This algorithm sometimes optimizes the primary objective at the expense of the secondary objective if the primary error tolerance requirement is very precise. To test the performance of the dynamic weighting mechanism with varying minimum error tolerances we have considered 2 cases of minimum error tolerances i) a strict case where $k_{tol} = 13.81$ and ii) a lenient case where $k_{tol} = 1$ (Fig. 3). These are two representative values chosen for comparison; any other values could also be chosen for testing.

B. Redirection position locations

Although in most cases the dissimilarities between the remote and local environments are not large, we wanted to test the performance of the dynamic weighted mechanism for redirections in all directions within one's physical reach. The complete positional map of all the locations tested is shown in Fig. 5. As can be seen, the radial distance of the positions is also varied to simulate interacting with near and far objects, with the outer radius of this positional map decided by the length of the arm kinematic chain. Each position is separated by 18 degrees in both horizontal and vertical planes. We have redirected the goal target to each of these locations. We test the redirections with both one and two goal target cases. In the case of two goal targets, the first target follows the same sequence as the one target case while for the second target a random sequence of target positions is selected from the target locations in Fig. 5. This same random sequence of target positions is used for all 2 target tests of different error tolerances and animation speeds. It is important to note that all redirections tested were performed with no locomotion, since if locomotion is involved then the optimal pose solution changes.

C. Sample animation speeds

We have tested the dynamic vs static weighting approach on four different speeds of motion capture animations of idling, walking, running, and sprinting. The rationale was to simulate varied speeds of action that users employ while performing tasks in remote-local MR/AR. For example, a sketching and painting task requires faster hand movements when compared to a book sorting task. These animations also have a varied range of body movements with sprinting having the most and idling having the least. The animation speed differences can be seen in the attached supplemental video.

D. Trends across various minimum error tolerances

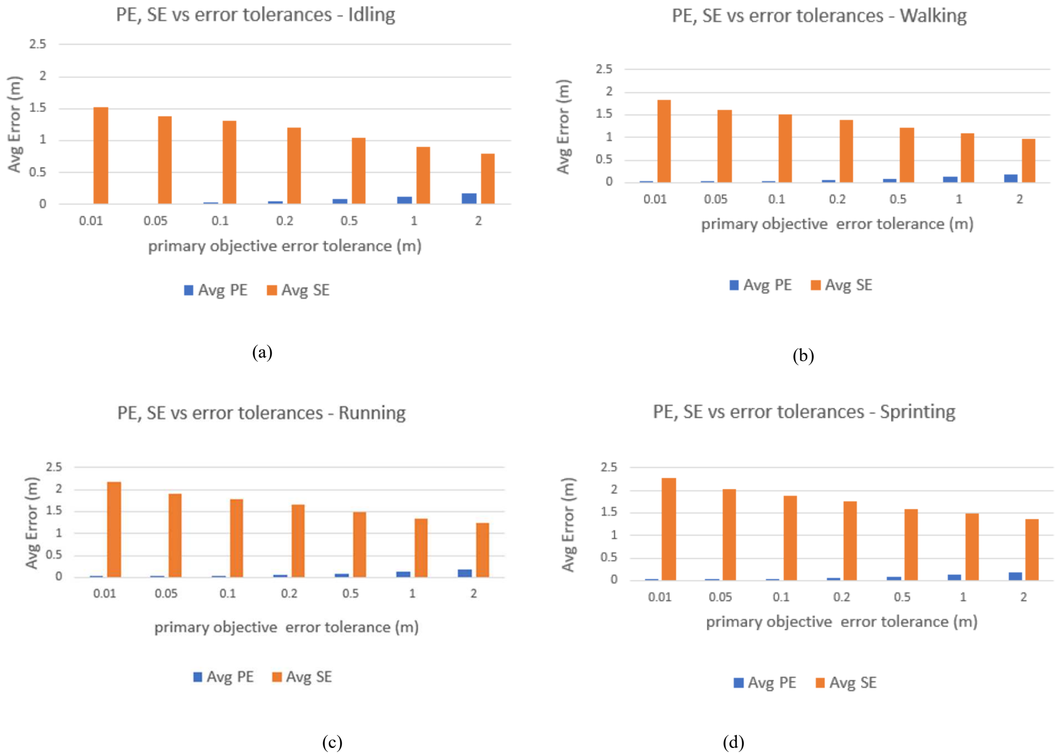
Our dynamic weighting approach allows the user to set the required minimum error tolerance. The weighting mechanism is designed in a manner to give full attention to the primary objective until the set error tolerance is met, after which it gives the remaining weight to the secondary objective. Hence, we were interested in seeing the trends in the primary and secondary errors as the minimum error tolerance values were eased. Different error tolerances were tested starting with 0.01m, 0.05m up to 2m.

VIII. RESULTS

We present the results in two sections: i) the performance of the static vs dynamic weighting approach, and ii) the trends of the primary and secondary errors using the dynamic weighting approach across different user-defined error tolerance values.

A. Performance of the dynamic vs static approach

The comparison results are presented in Tables I and II. They can be broadly divided into two categories of minimum



(a)

(b)

(c)

(d)

Fig. 6. Average (Avg) primary and secondary (PE and SE) trends for the dynamic weighted approach with the user defined minimum error tolerance values varied for (a) idling (b) walking (c) running (d) sprinting animations.

error tolerances i) strict case (Table I) and ii) lenient case (Table II). For each of these cases, we tested four different animation speeds with the remote-local workspace redirection locations as shown in Fig. 5. When reporting the % changes, we measure dynamic errors relative to the static errors. This is shown in Table I for the strict case where the primary error for the dynamic case has decreased by nearly 70 % (average of all animations and targets). However, the secondary error showed an increase of nearly 38% compared to the static case (average of all animations and targets). Similarly, Table II shows the average primary and secondary errors changes for the lenient case where the primary error for the dynamic case has decreased by nearly 20.5 % (average of all animations and targets) and the secondary error showed a very slight increase of 0.5 % compared to the static case (average of all animations and targets). The average secondary error is nearly double for faster animations of running and sprinting when compared to animations of idling and walking, though this is not because of the dynamic weighting scheme since it is seen in both the static and dynamic cases.

B. *Dynamic weighting trends across error tolerances*

The primary and secondary error trends for different animations across various error tolerance levels are shown in Fig. 6. We see the general expected trends of what the dynamic weighting mechanism is designed, i.e., as minimum error tolerance for the primary objective eases, the secondary objectives get optimized more resulting in a decrease in the average secondary error and an increase in the average primary error. It must be noted that the x-axis in Fig. 6 is not plotted on a linear scale. We also notice similar trends as in Tables I and II in Fig. 6 in terms of absolute values of the secondary error with slower animations of idling and walking having a smaller average error when compared to running and sprinting animations.

IX. Discussion and Limitations

The reduction in the net average error (primary objective error + secondary objective error) across various animation speeds and number of goal targets for the strict case is 34.4% and for the lenient case is 20.5 %. To put it in another way, compared to the net average dynamic error, the net average static error is 248% and 26% larger for the strict and lenient error tolerance cases, respectively. Higher net error reduction in the strict case is primarily due to the 72.97 % reduction in the primary objective error when compared to the lenient case where reduction in primary objective error is only 21%. The dynamic weighting approach can adapt across animation speeds and number of targets and we observe similar results for both one and two goal targets and different animation speeds for both the strict and lenient cases. Although the percentage errors across animation speeds between the dynamic and static cases, are similar, when comparing between the animations, faster animations of running and sprinting have higher body pose (secondary) error. This is not due to more primary objective optimization (in the dynamic case) since this trend is also seen for the static case, but because these animations have larger range of body movements causing increase of the distance error to a particular target during redirection. This can also be seen in the average secondary errors in Fig. 6.

One limitation of the study is that we have tested with motion captured human actions, which are smooth and have clean body motions. When capturing body pose angles with off-the-shelf RGBD cameras many factors may cause the data to be jittery. These include tracking problems, variable frame rates, and packet dropping during remote-local communication. Another limitation of the dynamic approach is the added computing requirement when compared to the static approach since the weights are calculated for every frame according to the current joint and goal configurations. This is an important factor considering that real-time joint redirection itself is a computationally intensive process. One solution to ease the computing requirement would be to calculate the dynamic weights once for a larger interval of frames. Although this would reduce the MO performance, it would not be perceptible to humans if the frame interval is not too large. Finally, even though the presented optimization framework includes human joint constraints, we have not evaluated the perception of the redirection using dynamic weighting for its naturalness and realness.

X. Conclusion and Future work

We presented a dynamic weight allocation approach for the MO problem of minimizing positional errors and preserving the pose of interactions in remote local MR/AR. The minimization of positional errors is considered as the primary objective in the optimization framework while pose preservation is the secondary objective. Prior work used a static method to distribute the weights among the primary and secondary objectives. These weights were set by the user on a trial and error basis. In our approach, the weighting is done dynamically considering the current joint and goal target configurations according to the minimum error tolerance which can be set by the user. We have tested our approach on motion captured human animations using different redirection positions, number of targets, and error tolerances. Our approach has shown a decrease in net average error ranging from 20.5% to 34.6 % when compared to the static approach. In the future, we plan to test the dynamic weighting approach on data collected when users are performing a task in mixed reality. We are also working on developing a framework to find optimal pose redirection for actions that also involve locomotion.

References

[1] Z. He, R. Du, and K. Perlin, “CollaboVR: A Reconfigurable Framework for Creative Collaboration in Virtual Reality,” *Proc. - 2020 IEEE Int. Symp. Mix. Augment. Reality, ISMAR 2020*, pp. 542–554, 2020, doi: 10.1109/ISMAR50242.2020.00082.

[2] D. Ungureanu *et al.*, “HoloLens 2 Research Mode as a Tool for Computer Vision Research,” *arXiv Prepr. arXiv2008.11239*, 2020.

[3] S.-E. Wei *et al.*, “Vr facial animation via multiview image translation,” *ACM Trans. Graph.*, vol. 38, no. 4, pp. 1–16, 2019.

[4] J. M. Amis and R. Greenwood, “Organisational change in a (post-) pandemic world: Rediscovering interests and values,” *J. Manag. Stud.*, vol. 58, no. 2, pp. 582–586, 2021.

[5] H. Joly, “Lead your team into a post-pandemic world,” *Harvard Bus. Rev. https//hbr. org/2020/05/lead-your-team-into-a-post-pandemic-world*, 2020.

[6] G. C. Kane, R. Nanda, A. Phillips, and J. Copulsky, “Redesigning the Post-Pandemic Workplace,” *MIT Sloan Manag. Rev.*, vol. 62, no. 3, pp. 12–14, 2021.

[7] B. Maturana, A. M. Salama, and A. McInneny, “Architecture, urbanism and health in a post-pandemic virtual world,” *Archnet-IJAR Int. J. Archit. Res.*, 2021.

[8] S. H. Mackenzie and J. Goodnow, “Adventure in the age of COVID-19: Embracing microadventures and locavism in a post-pandemic world,” *Leis. Sci.*, vol. 43, no. 1–2, pp. 62–69, 2020.

[9] J.-S. Lee and S.-H. Lee, “Automatic path generation for group dance performance using a genetic algorithm,” *Multimed. Tools Appl.*, vol. 78, no. 6, pp. 7517–7541, 2019.

[10] C. Beyan, V.-M. Katsageorgiou, and V. Murino, “Moving as a leader: Detecting emergent leadership in small groups using body pose,” in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 1425–1433.

[11] B. Yoon, H. Il Kim, G. A. Lee, M. Billinqhurst, and W. Woo, “The effect of avatar appearance on social presence in an augmented reality remote collaboration,” *26th IEEE Conf. Virtual Real. 3D User Interfaces, VR 2019 - Proc.*, pp. 547–556, 2019, doi: 10.1109/VR.2019.8797719.

[12] H. Hung and D. Gatica-Perez, “Estimating cohesion in small groups using audio-visual nonverbal behavior,” *IEEE Trans. Multimed.*, vol. 12, no. 6, pp. 563–575, 2010, doi: 10.1109/TMM.2010.2055233.

[13] D. Sanchez-Cortes, O. Aran, M. S. Mast, and D. Gatica-Perez, “A nonverbal behavior approach to identify emergent leaders in small groups,” *IEEE Trans. Multimed.*, vol. 14, no. 3 PART 2, pp. 816–832, 2012, doi: 10.1109/TMM.2011.2181941.

[14] S. Starke, N. Hendrich, and J. Zhang, “Memetic Evolution for Generic Full-Body Inverse Kinematics in Robotics and Animation,” *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 406–420, 2019, doi: 10.1109/TEVC.2018.2867601.

[15] C. Hoffmann, S. Büttner, M. Prilla, and K. Wundram, “Impact of augmented reality guidance for car repairs on novice users of AR: a field experiment on familiar and unfamiliar tasks,” in *Proceedings of the Conference on Mensch und Computer*, 2020, pp. 279–289.

[16] M. Norman, G. Lee, R. T. Smith, and M. Billinqhurs, “A mixed presence collaborative mixed reality system,” *26th IEEE Conf. Virtual Real. 3D User Interfaces, VR 2019 - Proc.*, pp. 1106–1107, 2019, doi: 10.1109/VR.2019.8797966.

[17] J. Young, T. Langlotz, M. Cook, S. Mills, and H. Regenbrecht, “Immersive Telepresence and Remote Collaboration using Mobile and Wearable Devices,” *IEEE Trans. Vis. Comput. Graph.*, vol. 25, no. 5, pp. 1908–1918, 2019, doi: 10.1109/TVCG.2019.2898737.

[18] D. Lindlbauer and A. D. Wilson, “Remixed reality: manipulating space and time in augmented reality,” in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 2018, pp. 1–13.

[19] J. Fan, A. Ullal, L. Beuscher, L. C. Mion, P. Newhouse, and N. Sarkar, “Field Testing of Ro-Tri, a Robot-Mediated Triadic Interaction for Older Adults,” *Int. J. Soc. Robot.*, pp. 1–17, 2021.

[20] J. Fan, L. C. Mion, L. Beuscher, A. Ullal, P. A. Newhouse, and N. Sarkar, “SAR-connect: a socially assistive robotic system to support activity and social engagement of older adults,” *IEEE Trans. Robot.*, 2021.

[21] K. Pfeuffer, M. J. Geiger, S. Prange, L. Mecke, D. Buschek, and F. Alt, “Behavioural biometrics in vr: Identifying people from body motion and relations in virtual reality,” in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019, pp. 1–12.

[22] T. Kim, A. Kachhara, and B. Maclntyre, “Redirected head gaze to support AR meetings distributed over heterogeneous environments,” *Proc. - IEEE Virtual Real.*, vol. 2016-July, pp. 207–208, 2016, doi: 10.1109/VR.2016.7504726.

[23] T. Piumsomboon, A. Dey, B. Ens, G. Lee, and M. Billinghurst, “CoVAR: Mixed-Platform Remote Collaborative Augmented and Virtual Realities System with Shared Collaboration Cues,” *Adjun. Proc. 2017 IEEE Int. Symp. Mix. Augment. Reality, ISMAR-Adjunct 2017*, pp. 218–219, 2017, doi: 10.1109/ISMAR-Adjunct.2017.72.

[24] R. Henrikson, T. Grossman, S. Trowbridge, D. Wigdor, and H. Benko, “Head-coupled kinematic template matching: A prediction model for ray pointing in vr,” in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020, pp. 1–14.

[25] K. Pfeuffer *et al.*, “ARtention: A design space for gaze-adaptive user interfaces in augmented reality,” *Comput. Graph.*, vol. 95, pp. 1–12, 2021.

[26] A. J. Cowell and K. M. Stanney, “Manipulation of non-verbal interaction style and demographic embodiment to increase anthropomorphic computer character credibility,” *Int. J. Hum. Comput. Stud.*, vol. 62, no. 2, pp. 281–306, 2005, doi: 10.1016/j.ijhcs.2004.11.008.

[27] T. Piumsomboon *et al.*, “Mini-me: An adaptive avatar for Mixed Reality remote collaboration,” *Conf. Hum. Factors Comput. Syst. - Proc.*, vol. 2018-April, pp. 1–13, 2018, doi: 10.1145/3173574.3173620.

[28] A. A. Canutescu and R. L. Dunbrack Jr, “Cyclic coordinate descent: A robotics algorithm for protein loop closure,” *Protein Sci.*, vol. 12, no. 5, pp. 963–972, 2003.

[29] A. Aristidou, Y. Chrysanthou, and J. Lasenby, “Extending FABRIK with model constraints,” *Comput. Animat. Virtual Worlds*, vol. 27, no. 1, pp. 35–57, 2016.

[30] C. Chen, Y. Zhuang, F. Nie, Y. Yang, F. Wu, and J. Xiao, “Learning a 3D human pose distance metric from geometric pose descriptor,” *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 11, pp. 1676–1689, 2010.

[31] Y. Liu, H. Ishibuchi, G. G. Yen, Y. Nojima, N. Masuyama, and Y. Han, “On the Normalization in Evolutionary Multi-Modal Multi-Objective Optimization,” in *2020 IEEE Congress on Evolutionary Computation (CEC)*, 2020, pp. 1–8.

[32] C. Shen, Y. Shi, and B. Buckham, “Path-Following Control of an AUV: A Multiobjective Model Predictive Control Approach,” *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 3, pp. 1334–1342, 2019, doi: 10.1109/TCST.2018.2789440.

[33] S. Menard, *Logistic regression: From introductory to advanced concepts and applications*. Sage, 2010.