XRgonomics: Facilitating the Creation of Ergonomic 3D Interfaces

João Belo
joaobelo@cs.au.dk
Aarhus University
Denmark

Anna Maria Feit
ETH Zürich
Switzerland
Saarland University, Saarland Informatics Campus
Germany

Tiare Feuchtner
Aarhus University
Denmark
Vienna University of Technology
Austria

Kaj Grønbæk
Aarhus University
Denmark

Figure 1: The XRgonomics toolkit aims to facilitate the design of ergonomic 3D UIs, common in mixed reality applications (left). We use a user's physiological model to compute the ergonomic cost of interaction at each reachable position in the interaction space (center). In XRgonomics, creators can visualize this cost through colored voxels in the interaction space: red indicates high and blue low-cost areas (right).

ABSTRACT

Arm discomfort is a common issue in Cross Reality applications involving prolonged mid-air interaction. Solving this problem is difficult because of the lack of tools and guidelines for 3D user interface design. Therefore, we propose a method to make existing ergonomic metrics available to creators during design by estimating the interaction cost at each reachable position in the user's environment. We present XRgonomics, a toolkit to visualize the interaction cost and make it available at runtime, allowing creators to identify UI positions that optimize users' comfort. Two scenarios show how the toolkit can support 3D UI design and dynamic adaptation of UIs based on spatial constraints. We present results from a walkthrough demonstration, which highlight the potential of XRgonomics to make ergonomics metrics accessible during the design and development of 3D UIs. Finally, we discuss how the toolkit may address design goals beyond ergonomics.

CHI '21, May 8–13, 2021, Yokohama, Japan

ACM ISBN 978-1-4503-8096-6/21/05.
https://doi.org/10.1145/3411764.3445349

CCS CONCEPTS

• **Human-centered computing** → **Mixed / augmented reality**; **Virtual reality**; **Systems and tools for interaction design**; *Accessibility design and evaluation methods*; *Walkthrough evaluations*; *Graphical user interfaces.*

KEYWORDS

3D User Interfaces, Ergonomics, Toolkit, Computational Interaction, Optimization, Adaptive User Interfaces, Mid-air Interaction

ACM Reference Format:
João Belo, Anna Maria Feit, Tiare Feuchtner, and Kaj Grønbæk. 2021. XRgonomics: Facilitating the Creation of Ergonomic 3D Interfaces. In *CHI Conference on Human Factors in Computing Systems (CHI '21), May 8–13, 2021, Yokohama, Japan.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3411764.3445349

1 INTRODUCTION

Cross Reality (XR) technologies are becoming mainstream as hardware gets more accessible, resulting in new applications across different sectors [10]. Despite the shift in interaction paradigms (e.g., from mouse input to mid-air interaction with controllers), interface elements and design guidelines for XR User Interfaces (UIs) are often inspired by 2D UI design. This influence can negatively affect user experience (UX) [28]. In particular, recent literature shows that creators struggle to address the physical aspects of XR experiences [1]. Existing challenges involve designing the posture of users and reducing fatigue. Remarkably, this problem persists even though substantial research in the HCI community has focused on mid-air interactions in the past decade, proposing design guidelines and evaluation metrics. A possible explanation is that these are difficult to apply during design and development of XR applications because:

- Proposed metrics [23] and models [25] focus on evaluating mid-air interactions that already exist but do not directly support the creation of new 3D UIs.
- General guidelines [3, 23] do not apply to the dynamic nature of MR applications that need to adapt constantly to the user's context [30].

To address these issues, we propose a method to make existing ergonomics metrics accessible to creators during design. We use a physiological model of the arm to assign a cost of interaction to any point in the user's reachable 3D space, that we call ergonomic cost. Its computation comprises the following steps:

(1) **Discretization of the interaction space** – transfer of the continuous interaction space into a discrete representation.
(2) **Computation of arm poses** – computation of multiple arm poses for each position in the interaction space using Inverse Kinematics (IK).
(3) **Computation of ergonomic cost** – calculation of the ergonomic cost for each arm pose using existing metrics and heuristics that assess ergonomics.

To make our method accessible to creators, we introduce XRgonomics - a toolkit to compute and visualize the ergonomic cost of the user's 3D interaction space. It comprises two major components: A Graphical User Interface (GUI) and an Application Programming Interface (API). The GUI allows creators to visualize the ergonomic cost associated with each position in the interaction space. The API gives access to this data at runtime to support development of adaptive interfaces. XRgonomics does not require any specifications about the XR application, making it easy to use during various design processes. To achieve this, we simplify the computation of the ergonomic cost by considering only static arm poses. We disregard users' arm motion between points of interaction which is difficult for creators to predict [1].

Two scenarios show how the toolkit can support the design of static UI elements and dynamic adaptation of UIs based on spatial constraints. To assess the usefulness of the toolkit, we present our findings from a walkthrough demonstration conducted with UI design experts. Finally, we discuss the potential of XRgonomics to address design goals beyond ergonomics. All the source code is available at: https://github.com/joaobelo92/xrgonomics.

2 BACKGROUND AND RELATED WORK

2.1 Designing for Ergonomics

2.1.1 Ergonomic factors in physical workstation design. Assessing ergonomic factors plays an important role when designing physical spaces, such as workstations, cars, and terminals. Much prior work in this domain estimates discomfort and ergonomic issues based on simple heuristics, such as joint angles. An example is RULA, a survey method for investigating work-related upper limb disorders [32]. RULA records working postures and attributes scores depending on risk factors. It assesses the risk for upper limb disorders considering aspects such as arm poses, movements, and forces.

Analysis of robot workspaces shares challenges also found in ergonomics design. For instance, Zacharias et al. [44] proposed a method to show which positions are easy to reach for robot arms. A significant challenge inherent to these scenarios is the limitation imposed by the agent's physical environment. For example, the physical space within a car cockpit limits the possibilities for where to mount a dashboard. In contrast, virtual workspaces are more flexible and allow the 3D user interface to adapt continuously to the user's context.

2.1.2 Ergonomic factors in mid-air interaction. Ergonomics are a significant factor in the design of virtual user interfaces, particularly in 3D UIs. Arm fatigue is one of the main issues designers must consider [28]. It is a common problem in interaction with vertical screens, also known as the *gorilla-arm effect* [7]. Researchers have proposed novel approaches to address this issue, ranging from novel interaction techniques [7, 15, 31] to UI optimization methods [33] to reduce muscle strain and fatigue. What these approaches have in common is that they intend to reduce fatigue in interaction. Among the most prominent qualitative methods to assess subjective fatigue are Likert scales [8], the NASA Task Load Index (NASA-TLX) [21], and the Borg CR10 scale [6]. HCI studies usually apply these approaches because they are non-invasive and do not require specialized equipment. However, substantial work must go into preparation and user studies, and these techniques provide only a coarse estimation of fatigue. While objective methods overcome some of these limitations, techniques used in biology and sports science often rely on external measurements, such as muscle activations [9], blood pressure [40], and heart rate [39]. Because these methods require specialized equipment and might interfere with the user's task, they are often inappropriate for HCI studies.

The HCI community has proposed alternatives to objective methods that are not intrusive. For example, Consumed Endurance (CE) [23] is a metric that tracks the user's arm pose to quantify arm-fatigue. CE computes the center of mass of the arm over time and uses that information to predict how long the user can continue

interaction before the shoulder muscles need rest. The authors show that the metric correlates well with the Borg CR10 scale and propose several guidelines for the design of mid-air UIs. Other studies use muscle activations from biomechanical models as indicators of fatigue. Bachynskyi et al. [2] show that predictions of muscle activation from static optimization correlate well with EMG data. In subsequent work, Bachynskyi et al. [3] applied biomechanical simulations to create a set of heuristics for designing 3D pointing interfaces, highlighting the potential of biomechanical simulations in UI design. Later, Jang et al. [25] proposed a method for modeling cumulative fatigue. Their approach quantifies arm fatigue by introducing a model for estimating muscle states (active, rest, fatigue) and uses a biomechanical arm model to estimate maximum shoulder strength. This approach makes it possible to consider periods of both interaction and rest.

Another ergonomic issue interlinked with fatigue is user comfort [28]. User comfort covers both physical and psychological dimensions, encompassing broader aspects such as posture and social awkwardness that may arise from using gestures in public spaces. Although there is work exploring subtle mid-air interaction [31], we are not aware of guidelines or objective metrics to evaluate this issue.

The primary goal of XRgonomics is to make ergonomics metrics accessible to creators during design. We use established objective metrics as heuristics for comfort, to assess the quality of positions in the interaction space regarding ergonomic factors, such as fatigue. Our toolkit supports several of the metrics introduced above, namely RULA, Consumed Endurance, and muscle activations.

2.2 Computational Support For UI Design

Already 20 years ago, Myers, Hudson and Pausch highlighted the need for toolkits to support the creation of user interfaces [34]. Since then, researchers have proposed several computational methods to support UI design (see survey by Oulasvista et al. for an overview [36]). Some of these methods focus on ensuring user performance, while others make suggestions to improve the aesthetic qualities of an interface. Such computational methods often differ in the degree of involvement of the designer. At one end of the spectrum, tools automatically create UI designs and do not require designer involvement [16]. Other toolkits support the creator by observing their design process, evaluating manually created solutions, and generating alternative designs or changes, which the creator can choose to follow [4, 42]. Studies show that such tools improve the quality of designs and inspire creators, ultimately resulting in a collaborative environment involving the designer and the toolkit [27].

The support of computational methods is crucial for MR applications where the context of the user continuously changes. Existing work has explored methods that automatically determine *where* to place virtual content [14, 17, 35]. Others have investigated *how* to display virtual content to the user [13, 26, 41], or a combination of multiple aspects [30]. However, none of these automated approaches considers ergonomics. Designing for the physical aspects of interaction is one of several common difficulties during the creation of XR applications, as highlighted recently by Ashtari et al [1]. Among the key challenges identified by the authors, we aim to address the lack of concrete and accessible design guidelines.

In this work, we use computational methods to support 3D UI design for MR and VR applications. XRgonomics is a toolkit that supports the visual exploration of the design space in terms of ergonomics, enabling creators to make informed decisions about where to place UI elements as part of their standard design process. Also, creators can use XRgonomics to guide the layout of 3D UIs at runtime and specify areas of interaction to avoid or prioritize.

3 ERGONOMIC COST PIPELINE

When designing XRgonomics, our primary goal was to create a method that supports the design of ergonomic user interfaces during the early design stages of XR applications. To facilitate accessibility, we did not want to impose constraints on the application itself, nor require the content creator to provide extensive input about the to-be-designed interface (e.g., usage data, user profiles, or physical environment). For that reason, we developed an approach that does not make assumptions about the interaction space or interaction techniques involved. Another noteworthy aspect is that interaction in XR applications is often context-dependent. Consider a typical Hololens 2[1] application, where the UI comprises virtual mid-air interaction with buttons and sliders. Contextual aspects such as the task, environment, or user's pose can limit interaction with the system. For this reason, general guidelines for ergonomic 3D interface design are often inappropriate for XR applications. To overcome this challenge, we facilitate exploration of the interaction space during UI design and allow developers to use ergonomics metrics at runtime. In our approach, we analyze the entire interaction space and assign a cost of interaction at each reachable position in 3D space. We call this the ergonomic cost. For it to be accessible in real-time, we propose a pipeline that shifts the computationally intensive tasks to a pre-processing stage. This ergonomic cost pipeline comprises three steps that we describe in the following sections. Our approach allows the comparison of distinct reachable positions regarding different ergonomic aspects, opening novel possibilities for designing and optimizing user interfaces.

3.1 Discretization of the Interaction Space

In the initial step of the pipeline, we transfer the continuous interaction space into a discrete representation. This is necessary to make the problem computationally tractable. Hence, we represent the interaction space as a 3D Cartesian grid and call each element a voxel - a common term in computer graphics. We define the interaction space based on the positions a human can reach and manipulate objects with his hands from a fixed torso position, a concept also known as the reach envelope [11]. We use a simple kinematic chain between the shoulder and hands. A user representation that includes both arms requires a fixed offset between the shoulders and thorax. However, the shoulder's mechanics are complex, and shoulder joint motion depends on its component joints [24]. Hence, this simplification results in some loss of precision, but not enough to justify a more complex kinematic chain for our use case.

To generate the interaction space's voxel representation, we start by setting up voxel dimensions with a default side length of 10cm.

[1] https://www.microsoft.com/en-us/hololens/hardware

Creators can adjust the voxels' side length to change the granularity of the interaction space representation. We use a simple algorithm that iterates through an overestimated 3D Cartesian grid in a cube (Figure 2, black cube). Its side length is equal to the kinematic chain dimensions, which delimit the arm's reach. Applications can include a calibration step, so these dimensions accurately reflect the user. In our standard implementation, we use the arm dimensions of the 50th percentile male [18]. Then, we verify which voxels belong to the interaction space, removing the voxels outside of the reach envelope (Figure 2, yellow sphere). We do this by checking whether the distance from the shoulder to the center of a voxel is smaller or equal to the user's arm length.

3.2 Computation of Arm Poses

At the end of the pipeline, the result will be the cost of interaction for each position in the user's reach. But first, we must compute multiple poses the arm can take to reach each voxel in the interaction space. Related work points out lower risks of injury and reduced muscle load for postures with the wrist in a neutral position [32] (deviation and twist is 0 degrees). We aim to find the pose that minimizes discomfort, and because postures of the wrist in neutral positions are considered optimal, we simplify the kinematic chain further by removing this degree of freedom. This results in a two-segment body of the arm, where the forearm and wrist constitute a single segment (see Figure 2, pink kinematic chain). While this approach considers fewer possible arm poses, it significantly reduces the complexity of the inverse kinematics (IK) process and computation time of the pipeline. We base our IK process on the work of Tolani et al [43]. Considering fixed end-effector and shoulder positions, the elbow is free to swivel on an axis between these two points (see Figure 3), allowing us to express the elbow position as a function of ϕ about the $\hat{\mathbf{u}}$ axis:

$$e = r[cos(\phi)\ \hat{\mathbf{u}} + sin(\phi)\ \hat{\mathbf{v}}] + c$$

Where r denotes the radius and c the center of the circle described by the swiveling elbow joint [43]. The variable ϕ controls the elbow position, which is at its lowest height when $\phi = 0$. Note that the arm's physiology constrains the ϕ value and we disregard unreasonable postures of the arm, based on impossible joint angles and elbow positions. Therefore, to generate arm poses, we increase ϕ by a constant value ψ, which determines how much the elbow rotates until it reaches an anatomically impossible threshold (e.g., 150 degrees). Here, the ψ value determines how fine-grained the discretization of the elbow position is. At this stage, it is possible to customize thresholds for ϕ and create additional rules, to consider factors such as a user's physical impairments or constraints imposed by hardware.

3.3 Computation of the Ergonomic Cost

Our toolkit implements established metrics from HCI and ergonomics research to assess the ergonomic cost of each reachable voxel. In theory, any metric that considers arm poses to assess ergonomic factors is appropriate to compute this ergonomic cost. XRgonomics currently supports consumed endurance (CE) [23], Rapid Upper Limb Assessment (RULA) [32], and muscle activations from biomechanical simulations. Notice that some of these metrics, such as CE, consider motion. In those cases, we adjust the metric to consider only static arm poses and use the result as a heuristic for strain. In other words, the ergonomic cost is a measure of how comfortable it is to maintain interaction at a specific position in the interaction space.

In the previous step of the pipeline, the toolkit generated several arm poses for reaching each voxel. We then compute the ergonomic cost for each of these poses, and assign the one of these costs (i.e., of the pose with least discomfort) to the corresponding voxel. We base this strategy on findings that humans tend to use more efficient poses [37].

In XRgonomics, a creator can compute the ergonomic cost using one of the supported metrics, or combine multiple metrics by assigning a weight to each. In the next sections we will describe how we applied each metric in our pipeline.

3.3.1 Consumed Endurance (CE). To quantify fatigue in mid-air interaction, CE [23] considers endurance of the shoulder in terms of torque as ratio to the interaction time. We follow the authors'

Figure 2: The interaction space is computed from an overestimated 3D Cartesian grid (black cube) and delimited by the user's reach envelope (yellow sphere). A simple kinematic chain representing the user's arm can be seen in pink.

Figure 3: Our inverse kinematics approach considers the shoulder and end-effector (i.e., hand) positions to be fixed, and the elbow is free to swivel about the shoulder-hand axis.

approach and use shoulder torque as an index for muscle strain. As the authors mention, when there is no motion, the shoulder torque has to match the gravity torque $\vec{g}$:

$$\vec{T}_{shoulder} = \|\vec{r} \times m\vec{g}\|$$

Where $\vec{r}$ is the distance from the shoulder joint to the center of mass of the arm, and m is the mass of the arm. Since we are working with static poses, we can directly compute the center of mass of each pose and apply the formula above. The result is a heuristic for the ergonomic cost based on the CE approach to estimate muscle contraction.

3.3.2 Rapid Upper Limb Assessment (RULA). RULA [32] assigns posture scores to the upper limbs, neck, trunk and legs, depending on joint angles. Combining that information with muscle and force scores, the method results in a final score to assess risk factors associated with upper-limb disorders. Even though our approach only considers arm poses, RULA posture ratings convey relevant information about postures that prevent or might result in upper limb disorders. Hence, we use RULA's posture scores to compute a score based on the joint angles of the upper and lower arms' joint angles (see posture scores for group A [32]). Low posture scores reflect a working posture with minimal risk factors, while higher numbers indicate an increasing presence of risk factors. The final score can indicate which positions in the interaction space are preferable to avoid upper-limb disorders.

3.3.3 Muscle activations from Biomechanical Simulations. Biomechanical simulations can estimate muscle activation for a motion, which can indicate energy consumption and fatigue [2]. Therefore, this method has great potential as a heuristic for the design of 3D UIs. Typical biomechanical simulation pipelines use experimental motion data, which typically involve mapping physical to virtual markers, scaling the model to match the subject dimensions, using inverse kinematics to compute joint angles, and a final step to estimate muscle activations [3]. For our simulations we use OpenSim 4.1[2], an open-source tool for biomechanical modeling and simulation [12], and the upper extremity model created by Saul et al. (MoBL) [38]. This model has the dimensions of the 50th percentile male, and must be scaled to support other arm dimensions. Because we generate arm poses in the previous step of the pipeline, we only use OpenSim to estimate muscle activations. However, we must convert our vector representation of the arm's pose into OpenSim's generalized model coordinates and generate corresponding motion files where each arm pose remains static over a short time (refer to the source code for more details). We use static optimization[3] to estimate muscle activations for each pose, which is a fast and efficient method. To run our simulations, we follow Hicks et al.'s recommendations [22]. We used reserve actuators to prevent the model from being under-actuated and avoid failures in static optimization. These reserve actuators complement the model's muscles when these cannot generate sufficient forces to achieve a pose. It is important that reserve moments are small or non-existent [22], so that the model's muscles exert most of the forces necessary to maintain each pose. Hence, we use low optimal forces in our reserve actuators, to ensure the cost function in the static optimization algorithm prioritizes muscle forces. Because MoBL is a complex model and static optimization can converge to different results, we analyze each pose over time and save the timeframe that minimizes reserve actuation for each pose. This results in an activation value for each muscle and reserve actuator in the model. To facilitate comparison with other metrics, we combine these into a single ergonomic cost value. To do so, we average the muscle activations and sum all the reserve actuators. To prioritize results that mostly use muscle forces, we penalize cases where reserve moments are high. Note that while muscle activation ranges from 0 to 1, the same does not apply to reserve actuators. Therefore, we set a threshold for the maximum acceptable reserve forces ($T_{reserve}$), based on the net joint moments [22]. Then, we divide the sum of the reserve forces by $T_{reserve}$, which will always result in a higher value than the average muscle activation, if it the reserve forces are above the threshold. This results in the following ergonomic cost function:

$$erg\ cost = \frac{\sum_{n=1}^{M} n_{activation}}{M} + \frac{\sum_{n=1}^{A} a_{activation}}{T_{reserve}}$$

Where M is the number of muscles and A the number of reserve actuators in the model.

4 THE XRGONOMICS TOOLKIT

The pipeline described in section 3 constitutes the central part of XRgonomics, a toolkit that gives creators of 3D applications easy

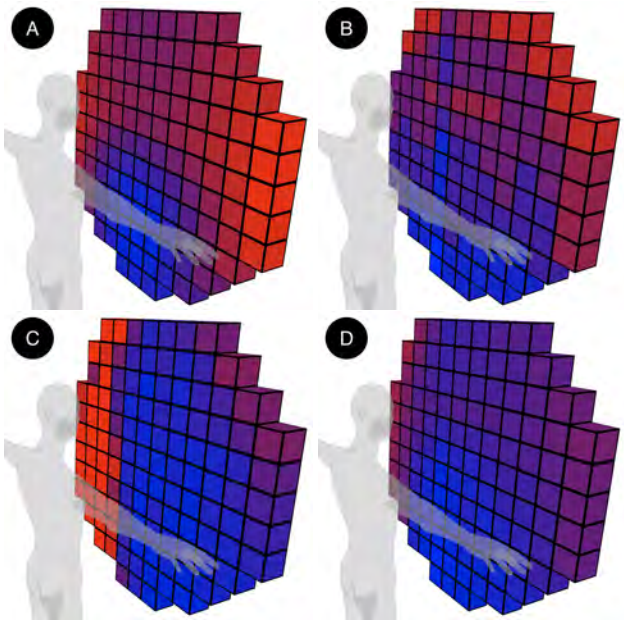

Figure 4: Visualization of the interaction space of the right arm using the supported metrics: A) Consumed endurance, B) RULA, C) Muscle activation, D) Weighted average (arithmetic mean in this case). The image shows only the voxels at the 40 cm slice (x-axis)

[2]https://simtk.org/projects/opensim
[3]https://simtk-confluence.stanford.edu/display/OpenSim/How+Static+Optimization+Works

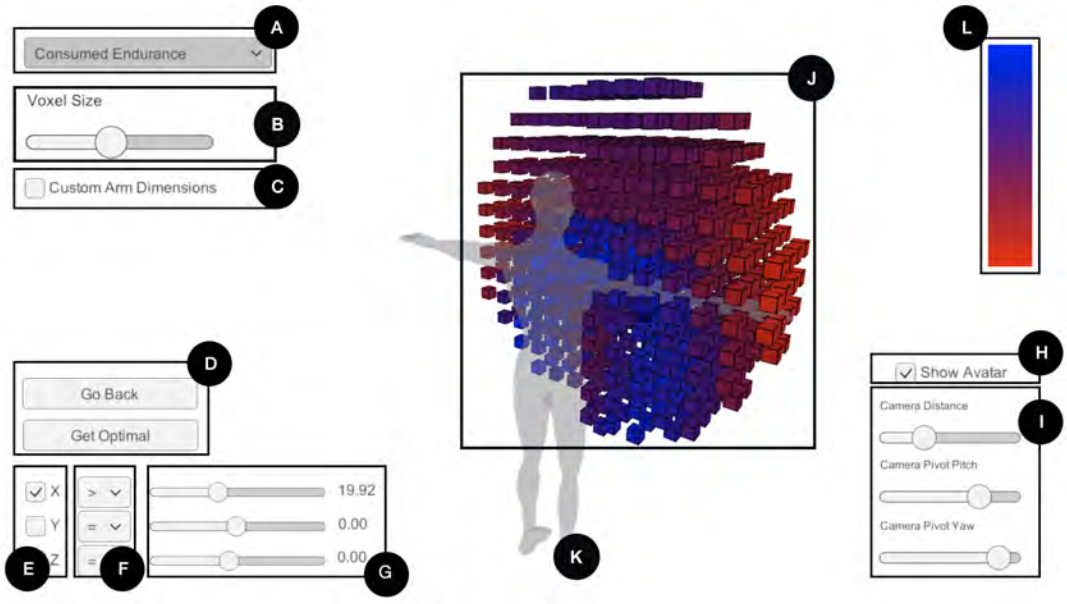

Figure 5: The XRgonomics GUI allows creators to visualize the interaction space and ergonomic cost of each voxel according to different ergonomic metrics. We will briefly describe each UI element: A) Dropdown menu for metric selection; B) Slider for voxel size setting; C) Menu to run computation pipeline for different arm dimensions; D) Buttons for retrieving the "optimal" voxel with the lowest ergonomic cost; E) Checkboxes for enabling/disabling spatial constraints; F) Dropdown list with comparison operators (=, >= or <=); G) Sliders for setting constraint values; H) Checkbox to toggle display of the avatar as visual reference for the shoulder position; I) Camera controls; J) Visualization of the interaction space and ergonomic cost in form of colored voxels; K) Avatar; L) Color mapping for the ergonomic cost, from blue (most comfortable) to red (least comfortable).

access to ergonomics metrics during design and development of 3D adaptive UIs. The toolkit comprises two major components: A Graphical User Interface (GUI) and an Application Programming Interface (API). The GUI allows creators to visualize the interaction space and each voxel's ergonomic cost. It can support the design of static interfaces (e.g., positioning virtual buttons on a desk) or give an overview of different metrics and their correspondent ergonomic cost for different positions in the interaction space (Figure 4). The API allows developers to use the ergonomic cost at runtime. This feature allows developers to create adaptive 3D UIs that consider user comfort as a criterion in the formulation of the optimization problem. For example, developers can retrieve voxels that minimize the ergonomic cost under specified spatial constraints in real-time. In this initial version of the toolkit, we consider only the right arm. Therefore, we set the center of the interaction space on the shoulder instead of the user's thorax. The source code for XRgonomics is available at https://github.com/joaobelo92/xrgonomics.

4.1 Graphical User Interface (GUI)

The GUI is implemented in Unity and uses the API to retrieve the ergonomic cost data. By default, it supports the arm dimensions of the 50th percentile male [18]. Creators can directly change parameters, such as the user's arm and voxel dimensions (Figure 5, C). Modifications to other parts of the pipeline require updates in the source code (see API section for more details). The GUI allows creators to visualize the interaction space and each voxel's ergonomic cost (Figure 5, J). The user can select between different ergonomic metrics supported by the toolkit (Figure 5, A). XRgonomics supports the metrics described in section 3.3 and a weighted average of those three metrics. Because the interaction space is a sphere, the voxels in the interior might be occluded. For that reason, the GUI features controls to apply spatial constraints on each coordinate axis (Figure 5, E), to limit the range of visible voxels. For example, it is possible to visualize a "slice" of voxels by adding an equality constraint on one axis (Figure 4). Creators can also reduce the rendering dimensions of the voxels to visualize data through more than one "slice" (Figure 5, B). An avatar is depicted in the center of the GUI, as a reference for the user's shoulder position in the interaction space (Figure 5, K).

Each voxel is colored according to the selected metric and the arm pose with the minimum ergonomic cost. As previously mentioned, we base this design choice on the principle that humans tend to use efficient poses [37]. Because CE and Biomechanical simulations output continuous results, we normalize all the ergonomic cost data using a simple feature scaling formula:

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

The color mapping is a linear interpolation from blue to red, representing low to high ergonomic cost, respectively (Figure 5, L). This mapping allows creators to visualize and compare voxels with similar values. Note that computed muscle activations from biomechanical simulations differ by small values when not influenced

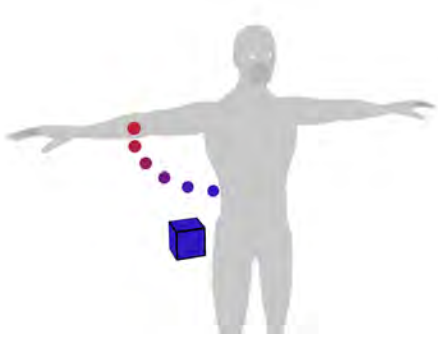

Figure 6: Creators can click on a voxel to visualize the possible positions of the elbow and the pose's ergonomic cost.

by reserve forces, which may result in identical voxel colors, even though there is a difference in average muscle activations. Therefore, we use a different normalization strategy to facilitate the visualization of this metric. We normalize the average muscle activations, multiply them by a scaling factor, and sum it with the ergonomic cost previously computed. This makes voxels with high reserve forces appear red, while smaller differences in the average muscle activations remain visible.

Finally, creators can click on a single voxel to visualize the arm poses generated by the IK process and their correspondent ergonomic cost (Figure 6). Since we are using a simplified kinematic chain, only the elbow positions differ in each pose.

4.2 Application Programming Interface (API)

We implemented the API in Python and used NumPy for most mathematical operations. The API has an endpoint to run the ergonomic cost pipeline for different arm and voxel dimensions, but developers must update the source code to change parameters in the inverse kinematics step, such as joint rotation limits or complex spatial constraints. While we implemented the algorithms for CE and RULA, the toolkit uses the OpenSim 4.1 Python bindings to run biomechanical simulations. However, XRgonomics does not directly support scaling of the arm or other changes to the biomechanical model, and the OpenSim tool is necessary for such tasks. We store voxel and ergonomic cost data in R*Trees [5], using an SQLite database. This makes it fast to find positions in 3D space that minimize a particular ergonomic metric or meet specific requirements. R*trees allow developers to query voxels within a bounding-box or arbitrary shapes like the area visible to a 3D camera. For networking, the API uses the ZeroMQ[4] framework, a fast messaging library. These networking features allow the API to communicate with the GUI (Unity), and enable developers to integrate XRgonomics in their applications. For example, developers can run the ergonomic cost pipeline for custom arm dimensions and retrieve data under specified spatial constraints. In our tests, the response time for such queries was less than 10ms, showing that the API can process requests in real-time and is suitable for XR applications.

[4]https://zeromq.org/

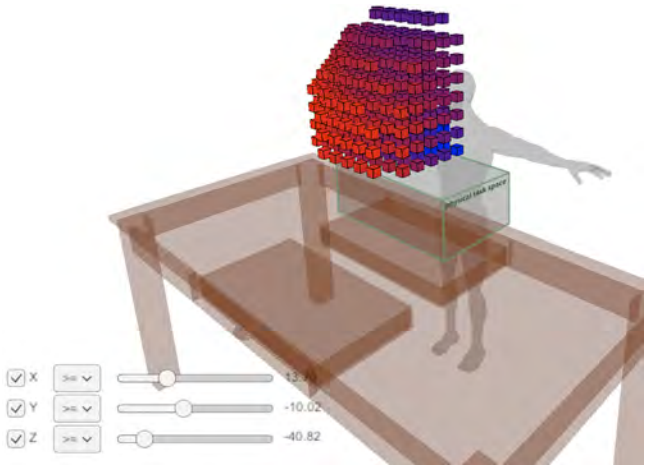

Figure 7: Creators can use the XRgonomics GUI to guide the design of static UI elements in traditional AR applications. Representations of physical/virtual objects can be added in the Unity scene to facilitate the task. In this case, the creators use constraints on the x and y axis to visualize the interaction space above and to the right zone of a table.

5 EVALUATION OF THE TOOLKIT

Ledo et al. introduced a categorization of evaluation strategies for HCI toolkit research [29]. We applied two strategies identified in this work to evaluate XRgonomics. First, we illustrate what the toolkit might support by discussing the usage of XRgonomics in two distinct scenarios: ergonomically optimized placement of static 3D UI elements, and runtime adaptation of a 3D UI based on dynamic constraints. Then, we collect feedback from potential toolkit users to explore its utility through a walkthrough demonstration.

5.1 Demonstration of Application Scenarios

To demonstrate the functionality of XRgonomics, we implemented two application scenarios for 3D UI design that we describe in the following sections:

5.1.1 Guiding the placement of static UI elements. Consider a "traditional" AR application, as Grubert describes it [19], where a designer defines the position of UI elements based on the user's pose. Creators can use the XRgonomics GUI to visualize the ergonomic cost for each position in the user's interaction space and guide the placement of 3D UI elements under specified constraints. For example, consider positioning virtual input elements on an office desk. The designer can analyze all the positions above the desk by setting constraints on different axes (see Figure 7), and use this information to design virtual elements such as a calculator or a drawing-board.

5.1.2 Dynamic adaptation of 3D UIs. Changes in the user's task, environment, and pose can limit interaction in MR applications. Because context changes are difficult or impossible to predict during design and development, a solution is to adapt the UI to the user's context at runtime. We implemented a prototype to show how creators can use XRgonomics to design adaptive ergonomic UIs. In this simplified MR scenario, we use the XRgonomics API to

Figure 8: A proof-of-concept application on the Hololens allows automatic placement of a virtual menu (music player) in the ergonomically optimal position within the user's FoV. The left image shows the user interacting with the virtual menu while looking straight ahead. The upper right visualization shows the ergonomic cost within the user's current FoV. When turning the head in other direction, the constraints are updated based on the new FoV (right picture). With a gesture, the user can summon the menu to reappear in the most comfortable position within this new zone of the interaction space.

adapt the placement of a virtual music player menu in a Hololens 2 application. The UI consists of a virtual 3D menu with buttons to play, stop, or change songs. Although the controls are easily accessible when the menu is visible, the limited field of view (FoV) of the Hololens can make interaction challenging. To overcome this issue, the user can request the menu to move into his FoV with a gesture (see Figure 8). The prototype then uses the XRgonomics API to identify the most comfortable and reachable position in the user's FoV and moves the virtual menu there. To achieve that, we use the view frustum of the Hololens as a spatial constraint. This application was implemented in Unity, using the Mixed Reality Toolkit (MRTK) and XRgonomics API. MRTK provides algorithms to facilitate the positioning of virtual menus (solvers[5]). However, these are limited to behavior like surface magnetism or following a virtual object and do not consider ergonomics.

5.2 Walkthrough Demonstration of Toolkit

To explore the utility of the toolkit, we conducted a walkthrough demonstration [29] with representatives from our target group, such as UI designers, developers, and HCI researchers. The study comprised six phases (Table 1). In each phase, we conducted semi-structured interviews with open-ended questions, rather than using for example questionnaires, to gain more in-depth insights. We will discuss the goals and findings from each phase in the following sections. Due to the COVID-19 pandemic, the study was conducted online through a video conferencing tool with screen sharing. For further reference, the study protocol, interview transcripts, and questionnaires are available in the project repository. We recruited eight participants (2 female; age: $M = 31.3, SD = 2.8$). All participants were familiar with UI design and XR technology, as they were professional software developers or VR/MR researchers. Most participants were uncertain about the concept of ergonomics, and none had prior knowledge of the metrics RULA, CE, or muscle activations.

[5]https://microsoft.github.io/MixedRealityToolkit-Unity/Documentation/README_Solver.html

5.2.1 Review of 3D UIs. In an initial discussion about 3D UI design, we aimed to learn about participants' prior knowledge and concerns regarding ergonomics. Three participants could relate directly to ergonomics and fatigue issues (P1, P3, P7) and all acknowledged the importance of the topic. However, none had addressed the problem in practice and they were not aware of existing metrics or strategies to use, apart from referring to guidelines for particular tools (e.g., ARKit[6] and ARCore[7]) (P3). These insights highlight one of the key barriers identified by Ashtari et al. [1], about the difficulties in designing for the physical aspects of AR/VR applications.

5.2.2 Introduction of GUI. In this phase, we introduced the XRgonomics prototype, explained the interaction space voxel representation, and instructed the participants on how to use the GUI (see section 4.1). To confirm that participants understood the visualization based on our explanations, we prompted them to describe the ergonomic characteristics of the interaction space when referring to a "slice" of voxels using the CE metric, as illustrated in Figure 5 (J). All participants showed an intuitive understanding of blue areas being "most comfortable" (P1), "easiest" (P5), and "most relaxed" (P7) to reach with the hand. We then challenged their understanding of this visualization, by pointing out some questionable CE results for positions above head-level (see Figure 5). Several participants expressed some uncertainty and even disagreement with the values in this area, which they perceived as hard to reach and therefore expected a higher ergonomic cost (e.g., P0, P3, P4, P7). However, instead of doubting the metric, they came up with likely explanations for why their opinions were wrong (e.g., P1, P3, P4, P5). We conclude that while the ergonomic cost and the toolkit visualization are easy to understand, creators might over-trust the tool, interpreting the visualizations as the ground-truth instead of reflecting on the validity of the metric. Hence, such tools should encourage creators to be critical and clarify the the metrics strengths and weaknesses.

[6]https://developer.apple.com/design/human-interface-guidelines/ios/system-capabilities/augmented-reality/
[7]https://designguidelines.withgoogle.com/ar-design/

1. Review of 3D UIs	2. Introduction of GUI	3. Design Task	4. Metrics overview	5. Introduction of API	6. Conclusion
Topic introduction and examination of prior knowledge	Instruction on how to use the GUI and visualization	Participants use the tool to design a static 3D UI with 3 elements	Demonstration and explanation of different ergonomics metrics	Demonstration and discussion of API features	General feedback and final remarks

Table 1: Overview of the study procedure consisting of a walkthrough demonstration of XRgonomics and a design task (phase 3), where participants used the GUI to create a static 3D UI.

5.2.3 Design Task. To evaluate the usefulness of XRgonomics in UI design, participants completed a design task using the toolkit. It consisted of planning the layout of three UI elements with different usability aspects (e.g., usage frequency) in a workstation (similar to Figure 5). At this stage, we enabled remote control of the mouse cursor, and the participants could use the toolkit running on the experimenter's PC. We asked them to think aloud while exploring the visualization, and show their desired UI element locations by pointing with their mouse or selecting a particular voxel. For a UI element that required frequent hand manipulation, all the participants used the toolkit to locate voxels with a low ergonomic cost. When deciding on the position for a rarely used element, with which inadvertent interaction is undesirable (e.g., "delete all"), participants pursued different strategies. To ensure the user makes a deliberate choice, some participants selected areas with a high ergonomic cost (P0-P3), while others also considered the workspace layout (P4-P7). All the participants stated that the visualization of the interaction space and ergonomic cost informed their decisions. Finally, when placing a non-interactive display element, participants pointed out that the supported metrics were not relevant, revealing an opportunity to integrate other metrics beyond ergonomics, such as visibility.

To explore the potential benefits of using XRgonomics in contrast to formulated guidelines from existing work, we quoted two design guidelines from the CE paper [23] and asked participants how they would apply these in the previous task. Participants highlighted several limitations of written guidelines, such as verbal statements being ambiguous or open to interpretation (P2-P6), whereas XRgonomics allows the designer to visually explore the interaction space (P0, P5, P7). Further, written guidelines may not apply if the recommended area is unavailable (e.g., because of physical restrictions). In contrast, setting constraints in XRgonomics allows the creator to analyze voxels in specific zones, compare, and identify locations that may not be the best overall but are optimal for a particular scenario (P0, P1, P3, P4).

5.2.4 Metrics overview. Next, we showed the ability to visualize different metrics in XRgonomics, briefly explaining the underlying theory and how the ergonomic cost is computed for CE, RULA, and muscle activation, respectively. All participants agreed that the visualizations aided their understanding of the underlying concepts, while appreciating that the toolkit makes the metrics accessible and useful without knowledge of the formal details.

5.2.5 Introduction of API. To explore the potential of XRgonomics to develop adaptive UIs, we explained the features accessible through the API and showed a video of the AR prototype described in section 4. Overall, participants appreciated the idea of generating constraints automatically and proposed several use cases for adaptive UIs. However, some participants mentioned that the toolkit should allow the designer or end-user to modify these constraints (P1, P2, P3) to address personal preferences, implicit spatial requirements, or a physical disability.

5.2.6 Conclusion. To collect general feedback and identify limitations of the toolkit, we concluded the study with some final questions. Participants agreed that the visualization provided understandable information about ergonomics in the interaction space, and mentioned that XRgonomics would help 3D UI design from early stages of design and development. They also proposed support for additional metrics beyond ergonomics, such as spatial relations between (physical/virtual) objects (P3, P5, P6), eye strain (P6), and visibility (P0, P1, P4-P7). A participant asked about having XRgonomics integrated into development tools, such as Unity (P4), which would facilitate access to it.

We conclude the results from this walkthrough demonstration highlight the potential of XRgonomics to make ergonomics metrics accessible during the design and development of 3D UIs.

6 DISCUSSION

In this work, we propose a method to estimate the ergonomic cost at each reachable position in the user's interaction space. We make this cost available to creators during design and development through XRgonomics, a toolkit to facilitate the creation of ergonomic 3D UIs. We demonstrated its potential through two examples: guidance for placement of static UI elements, and dynamic adaptation of 3D UIs optimized for comfort. Finally, we presented a walkthrough demonstration that highlights how XRgonomics can support UI design experts. We will now discuss the limitations of our approach, avenues for future work, and other relevant findings.

6.1 Limitations of XRgonomics

To create a method that runs in real-time, we simplified multiple steps of the pipeline. A simple kinematic chain limits the number of possible poses represented by the model to allow for a simple and fast inverse kinematics algorithm. Although in most cases a fixed wrist angle in the kinematic chain results in an ergonomic position, interaction with complex 3D input does not always work under such conditions, and environmental constraints might require poses with different wrist angles. Further, modeling the shoulder mechanism and its relation with the torso would require more complex IK.

Another design trade-off we made, was to ignore motion. Without context, existing models that analyze movement and fatigue are difficult to use, because it is hard to forecast certain aspects of interaction, like movement [1]. Therefore, we consider only static poses, allowing creators to easily use XRgonomics at design time.

Finally, XRgonomics currently supports metrics related to the ergonomics of the upper limbs. However, several other factors impact interaction in XR applications, such as visibility and consistency.

6.2 Future Work

Improvements to the IK implementation can result in higher accuracy and support more arm poses. In particular, extending the kinematic chain to consider the wrist angle is a natural improvement to the work we present. A possible approach would be to use a spiral point algorithm at each voxel to compute possible wrist positions. Another related improvement is to consider the user's torso position, with a model that represents the shoulder mechanism. This would result in more realistic arm poses and an improved representation of the interaction space. Another avenue for future work is to consider motion and model fatigue over time, considering the movement between voxels, their ergonomic cost, and muscle endurance. It will also be interesting to expand XRgonomics to consider other human factors beyond ergonomics of the upper limbs, such as vision, cognition, and spatial relations of objects.

Integration in existent MR and VR toolkits, such as MRTK solvers or Unity's IDE, is another important direction that would make our method more accessible to creators, as mentioned by study participants. The walkthrough demonstration also revealed several opportunities for GUI improvements, such as improved camera controls and better control for voxel selection (e.g., selecting between a range of values).

On a different topic, the user study revealed that participants' may over-trust the metrics when using the GUI. When we encouraged further reflection, participants reported doubts and treated the visualization more critically, after receiving explanations about how the metrics worked and their limitations. This highlights a potential issue of trust-calibration, which is an ongoing research topic in Visual Analytics [20]. To address this, XRgonomics could provide explanations for each metric and a disclaimer of their limitations.

6.3 Supported Ergonomics Metrics

In our current implementation, we incorporate three existing metrics to compute the ergonomic cost of interaction: CE, RULA, and muscle activations. While these represent important research in ergonomics, we discovered limitations throughout development and the user study. For instance, Hincápie-Ramos et al. proposed CE as a metric to quantify fatigue of mid-air interactions [23]. However, the main scenario considered in their work is interaction with vertical displays. We assume this influenced the design of the metric, which is base on the cross-product of the gravity vector and the center of mass of the arm for static poses. This results in questionable results when reaching overhead (Figure 4, A), which was a common topic of discussion in our study.

Then, RULA investigates risk factors associated with work-related disorders [32]. Although such information is relevant to the design of ergonomic 3D UIs, it does not consider poses where the arm is at rest. Moreover, it relies on wide-angle ranges for scoring arm poses, resulting in similar values for several voxels (Figure 4, B).

In our biomechanical simulations, the optimization algorithm did not always converge. Without inspecting each individual case, it was impossible to discern whether this was due to poses being physiologically impossible, or caused by issues with the model or optimization step. Nevertheless, we argue that biomechanical models can represent important information which other metrics cannot, such as physical constraints, muscles, and tendon length.

We believe that XRgonomics can support the understanding of existing ergonomic metrics and the development of new ones by offering a simple way to inspect, compare, and debug them.

7 CONCLUSION

In this paper, we proposed a method to estimate the ergonomic cost of interaction at each reachable position in the user's environment. We make it available through the XRgonomics toolkit, which aims to support the design of ergonomic 3D UIs by making existing ergonomics metrics accessible to creators. The GUI allows creators to visualize the user's reachable interaction space and the ergonomic cost in each position. The API allows the creation of complex and dynamic constraints, which enable real-time adaptation of 3D UIs (e.g., repositioning the UI to avoid hitting physical obstacles). We illustrate functionalities XRgonomics can support through two scenarios. Finally, a walkthrough demonstration of the prototype shows the usefulness of our approach and highlights its potential to integrate additional factors beyond ergonomics.

ACKNOWLEDGMENTS

We thank Aïna Linn Georges, Jens Emil Grønbæk and Hans-Jörg Schulz for their helpful feedback and discussions. Finally, we are grateful to Pedro Batalha for the illustrations, and to Sebastian Knudsen for the video. This work was supported by the Innovation Fund Denmark (IFD grant no. 6151-00006B), as part of the Manufacturing Academy of Denmark (MADE) Digital project, and by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant no. StG-2016-717054).

REFERENCES

[1] Narges Ashtari, Andrea Bunt, Joanna McGrenere, Michael Nebeling, and Parmit K. Chilana. 2020. Creating Augmented and Virtual Reality Applications: Current Practices, Challenges, and Opportunities. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3313831.3376722

[2] Myroslav Bachynskyi, Antti Oulasvirta, Gregorio Palmas, and Tino Weinkauf. 2014. Is Motion Capture-Based Biomechanical Simulation Valid for HCI Studies? Study and Implications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. Association for Computing Machinery, New York, NY, USA, 3215–3224. https://doi.org/10.1145/2556288.2557027

[3] Myroslav Bachynskyi, Gregorio Palmas, Antti Oulasvirta, and Tino Weinkauf. 2015. Informing the Design of Novel Input Methods with Muscle Coactivation Clustering. *ACM Trans. Comput.-Hum. Interact.* 21, 6, Article 30 (Jan. 2015), 25 pages. https://doi.org/10.1145/2687921

[4] Gilles Bailly, Antti Oulasvirta, Timo Kötzing, and Sabrina Hoppe. 2013. MenuOptimizer: Interactive Optimization of Menu Systems. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology* (St. Andrews, Scotland, United Kingdom) *(UIST '13)*. Association for Computing Machinery, New York, NY, USA, 331–342. https://doi.org/10.1145/2501988.2502024

[5] Norbert Beckmann, Hans-Peter Kriegel, Ralf Schneider, and Bernhard Seeger. 1990. The R*-Tree: An Efficient and Robust Access Method for Points and Rectangles. *SIGMOD Rec.* 19, 2 (May 1990), 322–331. https://doi.org/10.1145/93605.98741

[6] Gunnar Borg. 1990. Psychophysical scaling with applications in physical work and the perception of exertion. *Scandinavian Journal of Work, Environment & Health* 16 (1990), 55–58. http://www.jstor.org/stable/40965845

[7] Sebastian Boring, Marko Jurmu, and Andreas Butz. 2009. Scroll, Tilt or Move It: Using Mobile Phones to Continuously Control Pointers on Large Public Displays. In *Proceedings of the 21st Annual Conference of the Australian Computer-Human Interaction Special Interest Group: Design: Open 24/7* (Melbourne, Australia) *(OZCHI '09)*. Association for Computing Machinery, New York, NY, USA, 161–168. https://doi.org/10.1145/1738826.1738853

[8] James Carifio and Rocco J Perla. 2007. Ten common misunderstandings, misconceptions, persistent myths and urban legends about Likert scales and Likert response formats and their antidotes. *Journal of social sciences* 3, 3 (2007), 106–116.

[9] Mario Cifrek, Vladimir Medved, Stanko Tonković, and Saša Ostojić. 2009. Surface EMG based muscle fatigue evaluation in biomechanics. *Clinical Biomechanics* 24, 4 (2009), 327 – 340. https://doi.org/10.1016/j.clinbiomech.2009.01.010

[10] Perkins Coie. 2020. 2020 Augmented and Virtual Reality Survey Report. https://www.perkinscoie.com/en/ar-vr-survey-results/2020-augmented-and-virtual-reality-survey-results.html. Accessed: 2020-06-16.

[11] Joachim Deisinger, Ralf Breining, and Andreas Rößler. 2000. ERGONAUT: A Tool for Ergonomic Analyses in Virtual Environments. In *Virtual Environments 2000*, Jurriaan Mulder and Robert van Liere (Eds.). Springer Vienna, Vienna, 167–176.

[12] Scott L Delp, Frank C Anderson, Allison S Arnold, Peter Loan, Ayman Habib, Chand T John, Eran Guendelman, and Darryl G Thelen. 2007. OpenSim: Open-Source Software to Create and Analyze Dynamic Simulations of Movement. *IEEE Transactions on Biomedical Engineering* 54, 11 (2007), 1940–1950.

[13] Stephen DiVerdi, Tobias Hollerer, and Richard Schreyer. 2004. Level of detail interfaces. In *Third IEEE and ACM International Symposium on Mixed and Augmented Reality*. IEEE, New York, NY, USA, 300–301.

[14] Andreas Fender, Philipp Herholz, Marc Alexa, and Jörg Müller. 2018. OptiSpace: Automated Placement of Interactive 3D Projection Mapping Content. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3173574.3173843

[15] Tiare Feuchtner and Jörg Müller. 2018. Ownershift: Facilitating Overhead Interaction in Virtual Reality with an Ownership-Preserving Hand Space Shift. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology* (Berlin, Germany) *(UIST '18)*. Association for Computing Machinery, New York, NY, USA, 31–43. https://doi.org/10.1145/3242587.3242594

[16] Krzysztof Z. Gajos, Daniel S. Weld, and Jacob O. Wobbrock. 2010. Automatically generating personalized user interfaces with Supple. *Artificial Intelligence* 174, 12 (2010), 910 – 950.

[17] Ran Gal, Lior Shapira, Eyal Ofek, and Pushmeet Kohli. 2014. FLARE: Fast layout for augmented reality applications. In *2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, New York, NY, USA, 207–212.

[18] Claire C Gordon, Thomas Churchill, Charles E Clauser, Bruce Bradtmiller, and John T McConville. 1989. *Anthropometric survey of US army personnel: methods and summary statistics 1988*. Technical Report. Anthropology Research Project Inc Yellow Springs OH.

[19] Jens Grubert, Tobias Langlotz, Stefanie Zollmann, and Holger Regenbrecht. 2017. Towards Pervasive Augmented Reality: Context-Awareness in Augmented Reality. *IEEE Transactions on Visualization and Computer Graphics* 23, 6 (2017), 1706–1724.

[20] Wenkai Han and Hans-Jörg Schulz. 2020. Beyond Trust Building – Calibrating Trust in Visual Analytics. In *Proceedings of the Workshop on TRust and EXperience in Visual Analytics (TREX)*. IEEE, New York, NY, USA, 9–15. to appear.

[21] Sandra G. Hart and Lowell E. Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In *Human Mental Workload*, Peter A. Hancock and Najmedin Meshkati (Eds.). Advances in Psychology, Vol. 52. Elsevier (North Holland Publishing Co.), Amsterdam, Netherlands, 139 – 183. https://doi.org/10.1016/S0166-4115(08)62386-9

[22] Jennifer L. Hicks, Thomas K. Uchida, Ajay Seth, Apoorva Rajagopal, and Scott L. Delp. 2015. Is My Model Good Enough? Best Practices for Verification and Validation of Musculoskeletal Models and Simulations of Movement. *Journal of Biomechanical Engineering* 137, 2 (02 2015), 24. https://doi.org/10.1115/1.4029304 arXiv:https://asmedigitalcollection.asme.org/biomechanical/article-pdf/137/2/020905/6091748/bio_137_02_020905.pdf 020905.

[23] Juan David Hincapié-Ramos, Xiang Guo, Paymahn Moghadasian, and Pourang Irani. 2014. Consumed Endurance: A Metric to Quantify Arm Fatigue of Mid-Air Interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. Association for Computing Machinery, New York, NY, USA, 1063–1072. https://doi.org/10.1145/2556288.2557130

[24] Verne T Inman, JB deC M Saunders, and LeRoy C Abbott. 1944. Observations on the function of the shoulder joint. *The Journal of Bone and Joint Surgery* 26, 1 (1944), 1–30.

[25] Sujin Jang, Wolfgang Stuerzlinger, Satyajit Ambike, and Karthik Ramani. 2017. Modeling Cumulative Arm Fatigue in Mid-Air Interaction Based on Perceived Exertion and Kinetics of Arm Motion. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 3328–3339. https://doi.org/10.1145/3025453.3025523

[26] Simon Julier, Marco Lanzagorta, Yohan Baillot, Lawrence Rosenblum, Steven Feiner, Tobias Hollerer, and Sabrina Sestito. 2000. Information filtering for mobile augmented reality. In *Proceedings IEEE and ACM International Symposium on Augmented Reality (ISAR 2000)*. IEEE, New York, NY, USA, 3–11.

[27] Janin Koch, Andrés Lucero, Lena Hegemann, and Antti Oulasvirta. 2019. May AI? Design Ideation with Cooperative Contextual Bandits. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3290605.3300863

[28] Joseph J LaViola Jr, Ernst Kruijff, Ryan P McMahan, Doug Bowman, and Ivan P Poupyrev. 2017. *3D user interfaces: theory and practice*. Addison-Wesley Professional, Boston, MA, USA.

[29] David Ledo, Steven Houben, Jo Vermeulen, Nicolai Marquardt, Lora Oehlberg, and Saul Greenberg. 2018. Evaluation Strategies for HCI Toolkit Research. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, Article 36, 17 pages. https://doi.org/10.1145/3173574.3173610

[30] David Lindlbauer, Anna Maria Feit, and Otmar Hilliges. 2019. Context-Aware Online Adaptation of Mixed Reality Interfaces. In *Proceedings of the 32Nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) *(UIST '19)*. ACM, New York, NY, USA, 147–160. https://doi.org/10.1145/3332165.3347945

[31] Mingyu Liu, Mathieu Nancel, and Daniel Vogel. 2015. Gunslinger: Subtle Arms-down Mid-Air Interaction. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (Charlotte, NC, USA) *(UIST '15)*. Association for Computing Machinery, New York, NY, USA, 63–71. https://doi.org/10.1145/2807442.2807489

[32] Lynn McAtamney and E Nigel Corlett. 1993. RULA: a survey method for the investigation of work-related upper limb disorders. *Applied ergonomics* 24, 2 (1993), 91–99.

[33] Roberto A. Montano Murillo, Sriram Subramanian, and Diego Martinez Plasencia. 2017. Erg-O: Ergonomic Optimization of Immersive Virtual Environments. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology* (Québec City, QC, Canada) *(UIST '17)*. Association for Computing Machinery, New York, NY, USA, 759–771. https://doi.org/10.1145/3126594.3126605

[34] Brad Myers, Scott E. Hudson, and Randy Pausch. 2000. Past, Present, and Future of User Interface Software Tools. *ACM Trans. Comput.-Hum. Interact.* 7, 1 (March 2000), 3–28.

[35] Benjamin Nuernberger, Eyal Ofek, Hrvoje Benko, and Andrew D. Wilson. 2016. SnapToReality: Aligning Augmented Reality to the Real World. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 1233–1244. https://doi.org/10.1145/2858036.2858250

[36] Antti Oulasvirta, Niraj Ramesh Dayama, Morteza Shiripour, Maximilian John, and Andreas Karrenbauer. 2020. Combinatorial Optimization of Graphical User Interface Designs. *Proc. IEEE* 108, 3 (March 2020), 434–464.

[37] David A. Rosenbaum. 2010. Chapter 2 - Core Problems. In *Human Motor Control (Second Edition)* (second edition ed.), David A. Rosenbaum (Ed.). Academic Press, San Diego, 11 – 41. https://doi.org/10.1016/B978-0-12-374226-1.00002-4

[38] Katherine R. Saul, Xiao Hu, Craig M. Goehler, Meghan E. Vidt, Melissa Daly, Anca Velisar, and Wendy M. Murray. 2015. Benchmarking of dynamic simulation predictions in two software platforms using an upper limb musculoskeletal model. *Computer Methods in Biomechanics and Biomedical Engineering* 18, 13 (2015), 1445–1458.

[39] Suzanne C. Segerstrom and Lise Solberg Nes. 2007. Heart Rate Variability Reflects Self-Regulatory Strength, Effort, and Fatigue. *Psychological Science* 18, 3 (2007), 275–281. https://doi.org/10.1111/j.1467-9280.2007.01888.x PMID: 17444926.

[40] Gisela Sjøgaard, Gabrielle Savard, and Carsten Juel. 1988. Muscle blood flow during isometric activity and its relation to muscle fatigue. *European journal of applied physiology and occupational physiology* 57, 3 (1988), 327–335.

[41] Markus Tatzgern, Valeria Orso, Denis Kalkofen, Giulio Jacucci, Luciano Gamberini, and Dieter Schmalstieg. 2016. Adaptive information density for augmented reality displays. In *2016 IEEE Virtual Reality (VR)*. IEEE, New York, NY, USA, 83–92.

[42] Kashyap Todi, Daryl Weir, and Antti Oulasvirta. 2016. Sketchplore: Sketch and Explore with a Layout Optimiser. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems* (Brisbane, QLD, Australia) *(DIS '16)*. Association for Computing Machinery, New York, NY, USA, 543–555. https://doi.org/10.1145/2901790.2901817

[43] Deepak Tolani and Norman I Badler. 1996. Real-Time Inverse Kinematics of the Human Arm. *Presence: Teleoperators and Virtual Environments* 5, 4 (1996), 393–401. https://doi.org/10.1162/pres.1996.5.4.393

[44] Franziska Zacharias, Christoph Borst, and Gerd Hirzinger. 2007. Capturing robot workspace structure: representing robot capabilities. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, New York, NY, USA, 3229–3236.