Voice Snapping: Inclusive Speech Interaction Techniques for Creative Object Manipulation

Farkhandah Aziz
DMT Lab, Birmingham City University, Birmingham, United Kingdom, farkhandah.komal@mail.bcu.ac.uk

Chris Creed
DMT Lab, Birmingham City University, Birmingham, United Kingdom, chris.creed@bcu.ac.uk

Sayan Sarcar
DMT Lab, Birmingham City University, Birmingham, United Kingdom, sayan.sarcar@bcu.ac.uk

Maite Frutos-Pascual
DMT Lab, Birmingham City University, Birmingham, United Kingdom, maite.frutos@bcu.ac.uk

Ian Williams
DMT Lab, Birmingham City University, Birmingham, United Kingdom, ian.williams@bcu.ac.uk

Voice input holds significant potential to support people with physical impairments in producing creative visual design outputs, although it is unclear whether well-established interaction methods used for manipulating graphical assets within mainstream creative applications (typically operated via a mouse, keyboard, or touch input) also present benefits for speech interaction. We present three new voice controlled approaches utilizing interface snapping techniques for manipulating a graphical object's dimensions: NoSnap, UserSnap, and AutoSnap. A user evaluation with people who have physical impairments (*N*=25) found that each method enabled participants to successfully control a graphical object's size across a series of design tasks, although the automated snapping approach utilized within AutoSnap was found to be more efficient, accurate, and usable. Subjective feedback from participants also highlighted a strong preference for AutoSnap over the other techniques in terms of efficiency and ease of use.

CCS CONCEPTS • Human-centered computing ~ Accessibility ~ Accessibility design and evaluation methods

KEYWORDS: Voice Interaction, Inclusive Visual Design, Object Manipulation, Accessibility, Assistive Technology

1 Introduction

People with physical impairments who experience barriers and challenges in using traditional input devices (e.g. a mouse, and keyboard) can be excluded from using visual design applications such as Adobe Photoshop, Illustrator, XD, and Figma [1, 2]. Speech interaction holds potential to make these platforms more accessible [3, 4, 11], although there has been a lack of work to date investigating the feasibility of this approach. Initial research has explored the production of freeform creative drawings [5, 6, 21] and positioning of objects around a design canvas [30] (via voice control), although there remains little understanding around how fundamental graphical asset manipulations can be facilitated via speech. For instance, controlling an object's shape and dimensions is a core activity within creative applications where users can typically scale and resize objects (e.g. shapes and images) through dragging transformation handles via a mouse or touch input [22, 23, 24]. However, whilst this is a common feature that is widely used across different applications, it requires the use of dragging movements that do not clearly map to common speech interaction techniques. Similarly, object snapping is an important and relevant technique that is commonly used in creative design applications to support users with precise alignment and resizing of digital objects [8, 17, 18]. Snapping approaches (via mouse control) typically involve "smart" (sticky) snapping in alignment with other objects on the canvas or to guidelines that have been manually placed by users on the design canvas [7, 9]. Research has shown the use of snapping can support alignment and unity within designs [10, 24], thus supporting a designer's workflow and the production of professional outputs. However, similar to transformation handles, it is unclear whether object snapping can be beneficial where speech is the primary method of interaction to support people with physical impairments in manipulating creative objects.

We address the limited range of research in this area through developing and investigating new interaction techniques to support people with physical impairments in manipulating graphical assets within a digital canvas. In particular, we present three new voice controlled techniques which support object resizing manipulations via transformation handles and snapping – NoSnap, UserSnap, and AutoSnap. A user evaluation with participants who have physical impairments ($N = 25$) found all three approaches to be viable for manipulating a graphical object's dimensions, although AutoSnap was perceived to be more efficient, accurate, and usable than NoSnap and UserSnap. Subjective feedback also highlighted a strong preference for AutoSnap over the other two approaches in terms of efficiency and ease of use. This work therefore presents three primary contributions: (1) the development of new speech interaction approaches for resizing graphical assets informed through well-established object manipulation techniques, (2) a user evaluation with people who have physical impairments presenting new insights around the use of speech interaction for object manipulation (i.e. object resizing), and (3) research findings evidencing that automated object snapping for resizing actions (in voice control scenarios) presents interaction benefits in terms of usability and efficiency.

2 Related Work

2.1 Speech Interaction in Creative Work

Initial work has started to investigate the potential of speech interaction to support creative visual design and artistic work – for example, Harada et al. [3, 4, 5] explored the use of a vocal joystick controlled via vowel sounds to guide brush directions when completing freeform digital drawing work. Laput et al. [12] presented the PIXELTONE application where touch input was used to select parts of an image for editing operations such as applying filters and colors via voice commands. Similarly, Srinivasan et al. [13] used a combination of touch input and natural language commands where touch was used to specify the editing position on an image and natural language speech commands for image editing operations (e.g. "change fill color", "add a sepia filter"). Kim et al. [15] investigated the use of short vocal commands (e.g. "select", "crop", "brush", and "select & mask", etc.) in a creative context (i.e. Adobe Photoshop) and found short commands helpful for creative experts in reducing cognitive load when accessing various design features. Furthermore, Adobe XD [16] recently introduced grid numbers and labels for accessing application features via voice input (e.g. to select drawing tools, properties, layers). The application also supports positioning of the mouse cursor around the canvas via voice control (using commands such as "show grid", "drag from [grid number] to [grid number]", and "click [grid number]", etc.), although there is less emphasis on supporting object manipulation via voice control. This initial work highlights the broad potential of speech interaction to support and facilitate creative work, although there remains a lack of empirical research examining the optimal interaction techniques for fundamental visual design operations.

2.2 Speech Interaction for Object Manipulation

An essential and core element of visual design work is graphical object manipulation where digital assets can efficiently be positioned, scaled, resized, and adapted in terms of orientation [39]. Object positioning has been explored in the literature using speech input – for instance, Aziz et al. [30] investigated different speech supported interaction techniques for object positioning to assist people with physical impairments in creative design work (e.g. via the use of positional labels and alignment guides). Studies have also investigated multimodal speech input approaches for object manipulation – Hiyoshi and Shimazu [31] used the combination of mouse and speech input where mouse pointing was used for specifying a target position and voice commands for manipulating basic shapes (e.g. via statements such as "place the object here"). Elepfandt and Grund [42] explored the combination of voice and eye gaze interaction for object manipulation activities such as dragging and rotation of objects. Williams et al. [40, 41] also investigated the use of speech input in combination with hand gestures for 3D object manipulation (i.e. selection, deletion, position, rotation, and scaling). Moreover, Lee and Billinghurst [43] compared the performance of speech and gesture inputs for positioning 3D virtual objects within a digital space and highlighted the potential of using voice input for manipulating graphical objects. Whilst these studies demonstrate new opportunities around object manipulation via voice interaction, it remains unclear the extent to which common tools used in mainstream applications (e.g. the use of transformation handles, object snapping, etc.) can be utilised via speech controlled interfaces to facilitate more inclusive designer workflows.

2.3 Snapping and Alignment of Graphical Objects

Object dragging, alignment, and resizing actions in user interfaces are often assisted with "snapping" techniques using mouse or touch input [8, 20, 50]. Bier et al. [17] presented an early investigation using a snap dragging method to aid in precise alignment of objects. Similarly, Masui [18] introduced HyperSnapping which utilized a snapping grid to support users in aligning the position of a dragged object to other nearby digital assets (via mouse input). Dellisanti et al. [49] proposed a 3D object snapping approach to support the selection of objects within large displays. Baudisch et al. [7] also used a snapping approach to help users in aligning graphical assets (square shapes) based on surrounding objects. Furthermore, Xu et al. [24] presented a snapping method for alignment and equal spacing between design elements for enhancing the aesthetic appearance of an interface layout. Fernquist et al. [20] introduced "Oh Snap", a snapping technique which utilizes multiple snap points for aligning and positioning of graphical elements within touch interfaces. Van der Kamp [11] presented a multimodal approach which used eye gaze input for cursor pointing and voice control for drawing shapes which included a "snap" command to support creative workflow (although this was only used for initiating the design of basic shapes). Snapping features are commonplace in mainstream design applications via mouse control [44–47], although there is currently limited research on how this feature can potentially be utilized via speech-only interaction. It therefore currently remains unclear whether the benefits of snapping in more traditional interfaces can also be transferred over to voice controlled experiences for people with physical impairments.

3 Research Prototype

To address the limited research around manipulating an object's dimensions via speech input, we developed a web-based research prototype comprised of three different object manipulation techniques tailored for voice control. The prototype was developed using HTML, CSS, JavaScript (including the Web Speech API [25] for speech recognition) and presented a typical creative visual design interface (Figure 1). The design canvas (Fig. 1 (a)) contains a wireframe portfolio design mockup for a fictional professional designer comprised of common interface visual assets such as text and image placeholders of different sizes.

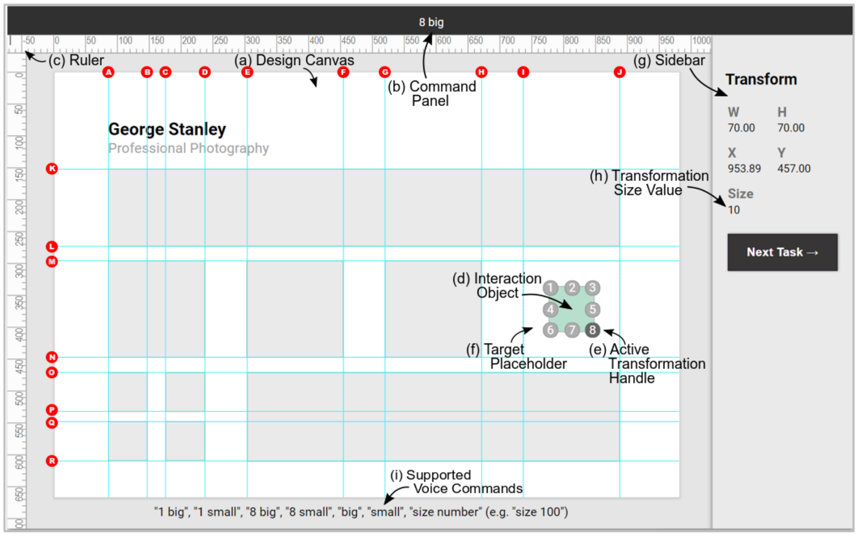


Figure 1. Main Research Prototype Interface – all guidelines (i.e. snap points) are displayed for reference, but are hidden by default

A speech command panel (Fig. 1 (b)) is displayed at the top of the screen as a black toolbar to help users visualize spoken voice commands which the system has recognized. A ruler (Fig. 1 (c)) is positioned at the top and left of the canvas with pixel values displayed on major ticks at 50 pixel intervals. An "interaction object" (Fig 1. (d)) is displayed as a green shape with eight transformation handles around its borders represented as circular labels with fixed numbers from 1 to 8. The interaction object can be resized by specifying the handle to be manipulated in combination with "big" or "small" utterances (e.g. "*x* big" or "*x* small" – where *x* is the number of a transformation handle from 1 to 8). The motivation for choosing "big" and "small" instead of other possibilities (such as "bigger", "smaller", "larger", and "shorter") was to ensure commands are short, quick, and easy to pronounce [15]. Once users issue a command, the object is resized in relation to the transformation size value (Fig 1. (h)) which can be altered through the voice command "size x" (where x relates to the number of pixels – e.g. "size 50", or "size 200", etc.). For instance, if the transformation size is set at "10" and the user issues a "4 big" command, the selected object will extend 10 pixels in width (from the left-side). This transformation occurs as a single "jump" from the current object size to the new size, as opposed to a continuous animation where the object is dynamically resized at a set speed. This decision was taken to avoid latency in processing speech recognition which can result in slight delays of commands being issued, thus leading to objects being resized beyond the user's intended target position [28, 38]. Target placeholders for a specific task (visualized using a white background with a dotted border) are displayed in relation to the interaction object (e.g. Fig. 1 (f)) and represent the final dimensions to which the object needs to be resized. A sidebar (Fig. 1 (g)) is used to display common object attributes such as width, height, *xy* positions, and transformation size. Supported voice commands (Fig. 1 (i)) are also available at the bottom of the design canvas to help users in recalling the available commands.

Switch input (e.g. a keyboard, mechanical switch, head tracker, foot pedal, etc.) is utilized for initiating the speech recognizer – audio feedback (a popping sound effect) is also played after a voice command has been issued to make the user aware that their input has been recognized. We developed three different object resizing approaches optimized for speech interaction: "NoSnap" (utilizing only transformation handles) and two object snapping techniques ("UserSnap" and "AutoSnap") which were focused around a common snapping feature in mainstream applications (i.e. Adobe XD and Figma). Figure 2 highlights this type of snapping approach where objects are resized through accessing a transformation handle via a mouse (or touch) and then dragging the object to the desired size. Whilst dragging, smart guides become visible which provide subtle visual hints for snapping the object in reference to other assets present on the canvas. Further details about each object resizing technique developed are provided in the sections below.

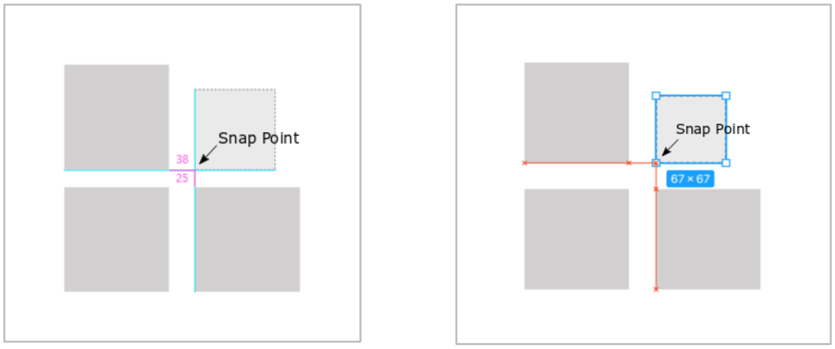
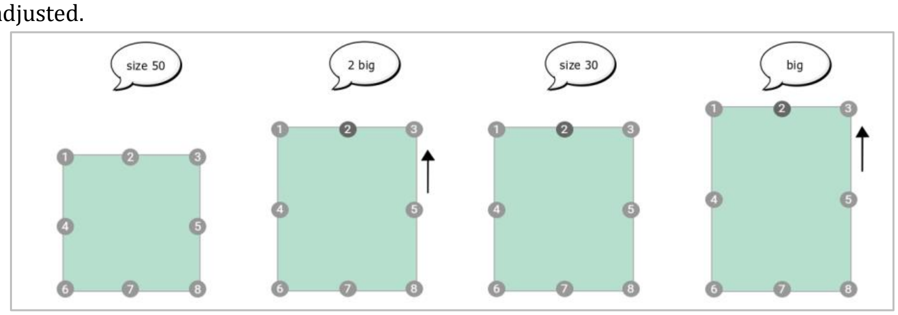


Figure 2. Smart snapping with Adobe XD (left) and Figma (right) – when the top-right object is resized via dragging and reaches the boundaries of other nearby canvas objects, smart snap guidelines are displayed

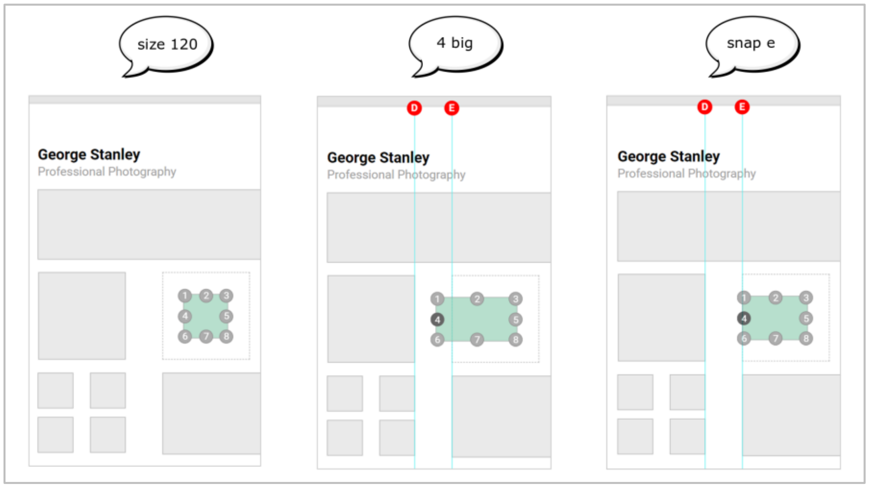
NoSnap: This approach uses voice commands to specify and manipulate an object handle (e.g. "1 big", "big", "8 small", "small", etc.), as well as enabling the control of transformation size through stating "size [number of pixels]" (e.g. "size 10"). Figure 3 illustrates an example where the top-side of a shape is extended – a "size 50" command is initially issued, followed by "2 big" to increase the object height by 50 pixels. The transformation size is then altered using a "size 30" command, followed by "big" to increase the object height by a further 30 pixels. A user can repeatedly issue the "big" or "small" command to continue manipulating a previously selected transformation handle. If a different selection handle is selected (e.g. "5 big"), the previous handle is deactivated and the new handle can then be adjusted.



(a) (b) (c) (d)

Figure 3. (a) Transformation size set to "size 50", (b) "2 big" resizes the object from the top side, (c) transformation size adjusted to 30 pixels via "size 30", (d) "big" command resizes the object height (from top-side)

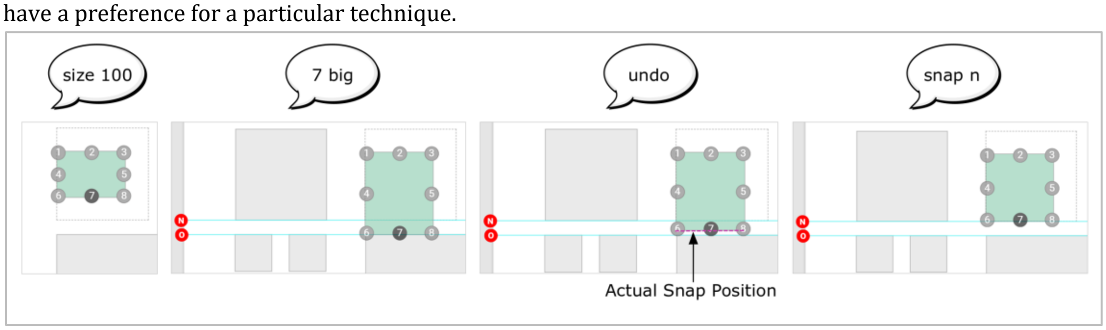
UserSnap: This approach combines the NoSnap features with object snapping in relation to nearby reference objects located on a digital canvas. A user can still resize an interaction object from any direction using the given voice commands ("1 big", "big", "8 small", "small" etc.), although a snap guide is displayed once the side of the object being manipulated is within a 100px threshold of a potential snap point. Each vertical and horizontal snap guideline is given a unique alphabetical identifier (A, B, C, etc.) displayed as a red circular label at the top and left edges of these guidelines. The user can then snap the object to the vertical or a horizontal guideline displayed using the voice command "snap x" (where x refers to the unique guideline identifier). The mock-up wireframe design consists of 10 vertical and 8 horizontal snap guidelines (Figure 1) – these are hidden by default and only guidelines within the 100 pixels threshold of the currently selected transformation handle are displayed. The threshold value was informed through previous research investigating mouse cursor snapping thresholds to support efficient target acquisition [19]. Figure 4 demonstrates how an interaction object can be resized via UserSnap.



(a) (b) (c)

Figure 4. UserSnap: (a) transformation size set to 120 pixels via "size 120" command, (b) user resizes object from left-side by issuing a "4 big" command, resulting in two guidelines "D" and "E" (within 100px threshold from left side of object) becoming visible (c) user issues "snap e" command to snap the object to their desired guideline

AutoSnap: This approach combines UserSnap with an additional automatic snapping feature where objects automatically resize to the closest available snap location (within a 100 pixel threshold). An "undo" voice command is also available to address scenarios where users do not require an automated snap – this results in the object being returned to its original size prior to the automatic snapping action (Figure 5). Users can then still utilize the "snap x" command (similar to UserSnap) to adjust the object's size to any available snap points. A potential advantage of AutoSnap is that it can make snapping actions more efficient through reducing the need for users to always have to state a vocal command to perform a snap (which is required in UserSnap). However, there is also the potential within AutoSnap for undesired resizing actions which could be tedious and frustrating users, whereas UserSnap provides full control over whether to perform a snapping action. We therefore wanted to investigate whether UserSnap and AutoSnap present any benefits when adjusting an object's dimensions and whether users have a preference for a particular technique.



(a) (b) (c) (d)

Figure 5. AutoSnap: (a) user sets the transformation size using command "size 100", (b) user resizes object from bottom through "7 big" command, resulting in two guidelines "N" and "O" becoming visible (within 100 pixels threshold) although the object is automatically snapped to closest available guideline "O", (c) user issues "undo" command to return to actual resize position, (d) user issues "snap n" command to snap object at desired guideline

4 User Evaluation

A user evaluation was conducted with people who have physical impairments to investigate the efficacy of NoSnap, UserSnap, and AutoSnap.

4.1 Participants

Twenty five participants with physical impairments (8 female) were recruited through online advertisements and existing network links. Participants were aged between 23 to 50 years (M=33.76, SD=6.94) and all were native English speakers. Table 1 details participant demographics, nature of physical impairments, and experience with interface prototyping, creative applications, speech technology, and assistive tools.

4.2 Procedure

Institutional Review Board approval was obtained for the study. All participants used their own computer and microphone for voice input, as well as their own chosen form of switch input for enabling the speech recognizer. Fifteen participants utilized a keyboard (i.e. the spacebar key), seven used dragon software [48] (e.g. via a vocal command such as “press spacebar”), two utilized a foot pedal, and one used a jellybean switch. The Google Chrome browser was required for experimental tasks to ensure browser compatibility with the Web Speech API [25]. Testing sessions were conducted online via Zoom [26] – the researcher initially provided participants with a link to the research prototype which they were asked to access and then share their screen content. After an overview of study focus was given by the researcher, participants were redirected to a consent page, followed by pre-test questions requesting details around demographic information and technical experience (in relation to interface prototyping applications and speech interaction). Participants were also asked about the nature of their impairments and any assistive tools they utilize. They were then asked to complete a training task which involved resizing a small interaction object to a larger size (highlighted through a target size placeholder) using relevant voice commands. An additional reference object was also provided for the UserSnap and AutoSnap practice tasks to ensure participants were able to familiarize themselves with the object snapping features.

After completion of the training task participants moved onto the main tasks for the first interaction approach they had been assigned to use (conditions and task order were counterbalanced to minimize order bias). There were ten object resize tasks for each interaction approach (i.e. 30 tasks in total) that involved adjusting the size of a green colored interaction object to the dimensions of a target placeholder (displayed as blank dotted box). At the start of a task, the interaction object was placed at the center of its corresponding target size placeholder to ensure that all sides of the object had to be manipulated in size (the position of the objects on the canvas could not be altered). A variety of interaction object and placeholder sizes were selected to ensure participants perform a range of different resizing tasks (Figure 6). These were informed though an analysis of standard interface elements within social media applications (profile covers, thumbnails, icons, logos, etc.) such as Facebook, LinkedIn, and Twitter. To initiate a task, participants activated the speech recognizer using their chosen form of switch input and then started to resize interaction objects as accurately as possible via the available speech commands and features. Participants continued with a task until they felt the interaction object’s dimensions had been accurately adjusted to the corresponding target placeholder’s size. The same process was repeated until all ten tasks for the condition had been attempted – a SUS form was then administered for participants to complete. After the same process had been completed for all three conditions, a semi-structured interview was conducted where participants were asked about what they liked and disliked about each interaction approach, their preferred method, overall impressions, and any suggestions for improvement. The testing session with each participant was video-recorded for later analysis. All testing sessions lasted between 50 minutes to 1 hour.

Table 1: Participants information: Physical impairments and condition details; IP = Interface Prototyping (Software); GM = Graphical Manipulation (Software); ST = Speech Technology; AT = Assistive Tools.

ID	*Age/ Gender*	*Physical Impairments*	*Condition Details*	*Technical Experience*
P1	42 (F)	Repetitive Strain Injury (RSI) Since (2016)	Difficulty in using fingers; Wrist Pain occasionally; Sometimes joint swelling and stiffness;	*IP*: Expert; *GM*: Expert; *ST*: Dragon software, Apple Siri; *AT*: Vertical mouse, Jellybean switch.
P2	35 (M)	RSI (Since 2010)	Hand tremors; Shooting pain in hands and arms; Pain in wrists; Tingling sensation in fingers.	*IP*: Average; *GM*: Average; *ST*: Dragon software, Google Assistant; *AT*: N/A.
P3	28 (M)	Tenosynovitis (Since 2020)	Wrist Pain; Joint swelling and stiffness; Difficulty in using fingers.	*IP*: Average; *GM*: Average; *ST*: Dragon; Apple Siri; *AT*: Head Tracker, Foot pedal.
P4	50 (F)	RSI (Since 2014)	Fatigue; Sore wrists occasionally; Shoulder pain; Pulsing pain in fingers.	*GD*: Average; *IP*: Average; *ST*: Google voice search services; *AT*: N/A.
P5	47 (M)	Motor Neuron Disease (Since 2016)	Muscle's weakness; Fatigue; Lack of balance; Unable to use hands	*IP*: Average; *GM*: Average; *ST*: Dragon software; Google Assistant *AT*: Tobii eye tracker.
P6	34 (M)	Muscular Myopathy (Since 2009)	Difficulty with walking without stick; Muscle's weakness; Fatigue; Lack of balance.	*IP*: Average; *GM*: Average; *ST*: Google speech services; *AT*: NA.
P7	26 (M)	Multiple Sclerosis (Since 2017)	Problem with balance; Tiredness; Numbness in fingers.	*IP*: Expert; *GM*: Expert; *ST*: Windows speech recognition; Google speech services, *AT*: NA.
P8	30 (M)	Tendinitis (Since 2015)	Fatigue; Pinched nerve; Muscle strains; Difficulty in holding stuff.	*IP*: Expert; *GM*: Expert; *ST*: Talon Voice, Google voice search; *AT*: Eye tracker.
P9	29 (F)	RSI (Since 2016)	Wrist pain, Pain in shoulders and upper arms; Tiredness; Stiffness in joints.	*IP*: Average; *GM*: Average; *ST*: Google Assistant, Samsung Bixby; *AT*: Head Tracker, USB Triple Foot Switch Pedal
P10	37 (M)	Lost Limb (Since 2018)	Amputated right arm	*IP*: Expert; *GM*: Expert; *ST*: Dragon software; *AT*: Foot pedal.
P11	36 (F)	RSI (Since 2012)	Weakness; Throbbing pain effect on hands occasionally; shoulders pain; Sometimes joint swelling at wrist.	*IP*: Average; *GM*: Average; *ST*: Apple Siri, Google voice search; *AT*: Jellybean Switch.
P12	29 (M)	RSI (Since 2015)	Discomfort in hands; Pain in fingers; Tiredness in arms;	*IP*: Average; *GM*: Expert; *ST*: Dragon software, Amazon Alexa; *AT*: NA.
P13	30 (M)	Spinal Muscular Atrophy (Type 2) (Since 1999)	Uses powered chair; Cannot walk since age 3; Unable to move hands and legs; Muscle's weakness; Lack of balance.	*IP*: Average; *GM*: Average; *ST*: Talon voice, Dragon software, Google Assistant; *AT*: Eye tracker, Head Pointer.
P14	23 (M)	RSI (Since 2017)	Shooting pain in hands and arms; Hand tremors occasionally; Tingling; Pain in wrists.	*IP*: Average; *GM*: Average; *ST*: Google speech services; *AT*: NA.
P15	26 (F)	RSI (Since 2018)	Aching fingers; weakness in hands and arms muscles; Numbness in fingers; painful wrists.	*IP*: Average; *GM*: Expert; *ST*: Mac voice control, Google Assistant; *AT*: NA.
P16	33 (M)	Multiple Sclerosis (Since 2012)	Fatigues; Numbness in arms and legs; Clumsiness; Lack of balance.	*IP*: Average; *GM*: Average; *ST*: Amazon Alexa, Apple Siri; *AT*: NA.
P17	42 (F)	Motor Neuron Disease (MND) (Since 2017)	Uses walking stick; Arms and shoulders pain; Fatigue.	*IP*: Average; *GM*: Average; *ST*: Windows speech recognition, Google speech services; *AT*: Head Tracker, Eye Tracker.
P18	29 (M)	RSI (Since 2017)	Hand tremors; Shooting pain in hands and arms; Pain in wrists; muscle weakness.	*IP*: Average; *GM*: Expert; *ST*: Google Home, Dragon software; *AT*: NA.
P19	35 (M)	RSI (Since 2006)	Shoulder pain; tiredness in forearms; Sore wrists occasionally; Pulsing pain in fingers.	*IP*: Expert; *GM*: Expert; *ST*: Google speech services, Mac voice control; *AT*: Foot pedal.
P20	40 (F)	Motor Neuron Disease MND (Since 2018)	Weak grip, Hard to climb stairs, Weak muscles	*IP*: Average; *GM*: Expert; *ST*: Google speech services; *AT*: NA
P21	28 (M)	Shoulder Impingement Syndrome (2020)	Weakness in arms, Pain in shoulders, Severe pain when lift arms above head	*IP*: Average; *GM*: Average; *ST*: Dragon software, Apple Siri; *AT*: NA.
P22	38 (M)	RSI (Since 2018)	Pain in forearms and elbows; Throbbing sensation in fingers; joint swelling sometimes.	*IP*: Average; *GM*: Expert; *ST*: Windows speech recognition; *AT*: NA.
P23	34 (F)	RSI (Since 2011)	Occasionally severe pain in hands; Tiredness in shoulders and upper arms;	*IP*: Average; *GM*: Average; *ST*: Samsung Bixby, Google Assistant; *AT*: NA.
P24	23 (M)	RSI (Since 2017)	Stiffness of joints; feeling of numbness in fingers; muscles weakness;	*IP*: Expert; *GM*: Expert; *ST*: Dragon, Google Assistant; *AT*: Trackball mouse.
P25	40 (M)	RSI (Since 2005)	Fatigue; Shoulder pain; sore wrist; sometimes throbbing pain in hands and fingers	*IP*: Average; *GM*: Expert; *ST*: Dragon software, Talon voice; Apple Siri; *AT*: NA.

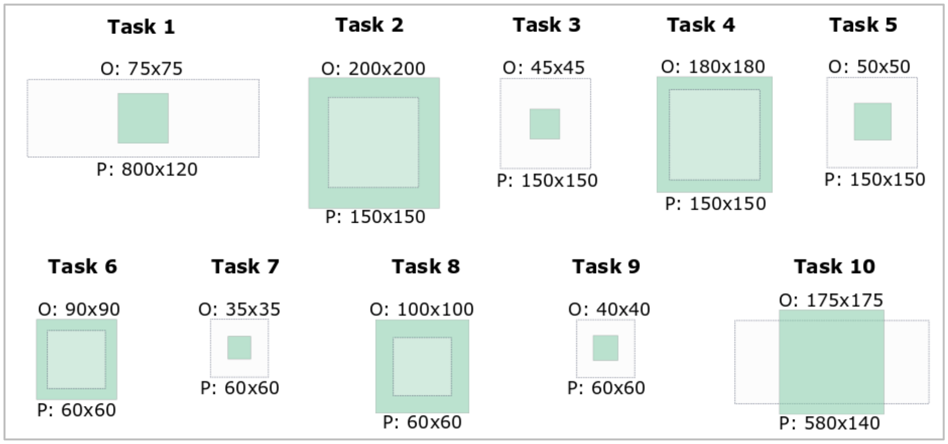


Figure 6. Interaction Object (O) and Placeholder sizes (P) for experimental tasks

4.3 Measures

Task completion time (measured in milliseconds), resize accuracy, speech recognition performance, and SUS scores [27] were calculated to evaluate the three interaction techniques. Resize accuracy was measured through ascertaining differences between the final width and height of interaction objects and target placeholder dimensions, and then calculating the average of Euclidean distances [32]. Speech recognition performance was categorized into three areas: "Speech Misrecognition" – where the speech recognizer incorrectly identified voice commands (e.g. "big" was identified as "dig"), "System Error" – where the system did not perform an action due to latency issues with the Web Speech API, and "Unsupported commands" – where users stated vocal input unrelated to the available system commands. System Usability Scale (SUS) was used to evaluate perceptions of usability for each approach.

5 Results

The Shapiro-Wilk's [29] normality test found task completion, resize accuracy, speech performance, and SUS scores were not normally distributed. Hence, we used non-parametric Friedman test of differences for repeated measures with Bonferroni correction for analysis. Wilcoxon signed rank [33] was used as post-hoc test to analyse the differences in task completion time, resize accuracy, speech performance, and SUS scores.

5.1 Task Completion Time

Average task completion time for NoSnap was 12.11 (SD=1.71), UserSnap 12.64 (SD=1.69), and AutoSnap 9.09 (SD=1.03). Friedman test results highlighted significant differences in task completion time (X^2=0.001, df=2, p<0.05). Post-hoc Wilcoxon signed rank showed a significant difference in task completion time between NoSnap and AutoSnap (Z=-4.37, p<0.001) and UserSnap and AutoSnap (Z=-4.37, p<0.001). However, no significant differences were observed between NoSnap and UserSnap (Z=-1.87, p=0.061). Figure 7 shows the average task completion time (in minutes) across the three interaction approaches.

5.2 Resize Accuracy

Average resize accuracy based on average Euclidean distance values for NoSnap was 0.84 (SD=0.81), UserSnap 0.32 (SD=0.47), and AutoSnap 0.24 (SD=0.46). Friedman test results highlighted significant differences in resize accuracies (X^2=0.001, df=2, p<0.05). The post-hoc Wilcoxon signed rank showed a significant difference in resize accuracy between AutoSnap and NoSnap (Z=-8.39, p<0.001), as well as UserSnap and NoSnap (Z=-7.45, p<0.001).

No significant differences were found between AutoSnap and UserSnap (Z= -1.81, p=0.70). Figure 8 represents the average resize accuracy across the three interaction approaches.

5.3 Speech Performance

The total number of vocal commands issued across all 25 participants for NoSnap were 3591 (SD=11.31), 4072 (SD=10.06) for UserSnap, and 2986 (SD=10.29) for AutoSnap. There were 182 (5.07%) "Speech Misrecognition" errors for NoSnap, 219 (5.38%) for UserSnap, and 150 (5.02%) for AutoSnap. Friedman test results showed no statistically significant differences for "Speech Misrecognition" across all three methods (X^2=0.31, df=2, and p>0.05). In terms of "System Errors", 73 (2.03%) commands were related to NoSnap, 87 (2.14%) for UserSnap, and 56 (1.87%) for AutoSnap. Friedman test results again showed no statistically significant differences for "System Errors" across all three methods (X^2=0.20, df=2, and p>0.05). There were 6 (0.17%) "Unsupported Commands" issued in NoSnap, 11 (0.27%) for UserSnap, and 9 (0.30%) for AutoSnap. Friedman test results found no statistically significant differences for "Unsupported Commands" across all three methods (X^2=0.30, df=2, and p>0.05).

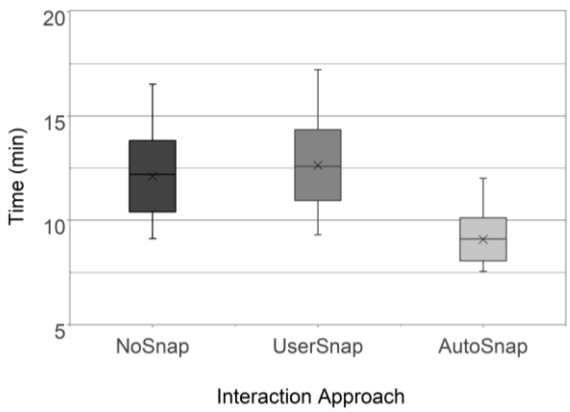


Figure 7. Average Task Completion Time

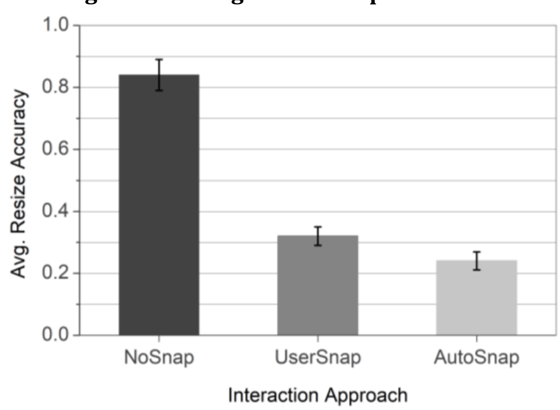


Figure 8. Average Resize Accuracies

5.4 Usability Scores

The average SUS score for NoSnap was 70.40 (SD=3.72), 73.20 (SD=11.42) for UserSnap, and 82.00 (SD=3.75) for AutoSnap. NoSnap and UserSnap scores can therefore be labelled as exhibiting a "Good" level of usability, while AutoSnap can be labelled as "Excellent" [27]. Significant differences were found across the three interaction approaches using Friedman test (X^2=0.001, df=2, p<0.05). The post-hoc Wilcoxon signed rank also found a significant difference between NoSnap and AutoSnap (Z=-4.30, p<0.001), NoSnap and UserSnap (Z=-2.10, p<0.001), and AutoSnap and UserSnap (Z=-3.43, p<0.001).

5.5 Qualitative Feedback

A thematic analysis [51] was conducted across all video recordings and responses to interview questions with several key themes identified.

User Perceptions of Voice Snapping Techniques: All participants provided positive feedback in terms of using the three different object resizing techniques. Sixteen participants preferred AutoSnap, seven preferred UserSnap, and two preferred NoSnap. Positive comments in relation to AutoSnap highlighted that it was "*intuitive*", "*reliable*", "*saves time*", and "*reduces the use of voice commands*" over the other methods. In particular, thirteen participants stated that the reduction of commands in AutoSnap could help to avoid vocal strain in comparison to UserSnap (where snapping commands always have to be issued). P16 stated that AutoSnap is more helpful for resizing square shaped objects with matching dimensions (e.g. 50x50 or 100x100), although it is not as efficient as UserSnap for objects where width and height may differ (e.g. 800x120, 400x200, etc.). For instance, when using corner transformation handles (i.e. 1, 3, 6, 8) in AutoSnap for objects with different dimensions (i.e. 800x120), two snapping actions could occur to both horizontal and vertical snap points simultaneously (which may be undesired). Seven participants therefore directly stated that UserSnap was more effective in this scenario as it provided more control over resizing actions ("*I like UserSnap as it works consistently for resizing object from all sides so I felt more control over transformation when using this method*" [P12]).

Voice Commands: There were no major speech recognition issues identified and all participants were able to successfully complete all object resize tasks. We observed there were range of misrecognized voice commands such as the "big" command being identified as different homophonic words such as "dig" or "wig", while "2" was identified as "too", and "4" as "for". It was also found that participants used directional commands on occasions such as "left big" instead of "4 big", "go down" for "2 small", and "right right" for "5 big". One participant also used a "5 large" command as opposed to using "5 big" and "left shorter" for "4 small". Moreover, two participants attempted to chain commands together using a combination of the size command coupled with transformation size (e.g. "4 big 10" with an expectation that it would complete two actions ("4 big" and "size 10")).

Transformation Size Estimation: In relation to NoSnap, eight participants emphasised an issue around estimating the size of transformations to efficiently complete a resize task (e.g. "… *It is hard to estimate correct transformation size value in first attempt so I tried to use ruler but then I had to calculate distance between ruler values to get correct transformation size value which required effort"* [P15]). Similarly, P11 highlighted that the *"… ruler helped to get bigger transformation size value … but it is hard to estimate correct value when you need to make small adjustments*". This was a common theme with NoSnap, although it was not highlighted in relation to UserSnap and AutoSnap as a significant challenge.

6 Discussion and Future Work

This paper has presented new voice controlled snapping techniques for manipulating the size of graphical objects within a digital design canvas. Participants with physical impairments found all three approaches to be viable and usable, although AutoSnap was perceived to be more efficient, accurate, and usable than both NoSnap and UserSnap. Subjective feedback from participants also correlated with quantitative findings with participants providing positive comments around the efficiency and intuitive nature of the snapping approaches (over NoSnap). Moreover, the clear preference for the UserSnap and AutoSnap techniques highlights that the snapping features developed were beneficial in the context of the object manipulation tasks that participants completed. This work therefore contributes a deeper understanding around the feasibility of voice controlled snapping approaches to support people with physical impairments when completing digital creative tasks.

In particular, the results indicate that common snapping techniques operated via traditional input devices [7, 17, 18] can be effectively tailored for speech interaction (within a creative context), thus building on initial work in this area [11]. The voice snapping approaches developed may also present wider benefits in terms of other common object manipulation activities such as positioning and rotation of graphical objects on a design canvas [23, 31, 39], simultaneous transformation of multiple objects (e.g. selecting multiple images and positioning them against a snap point), and approaches for even distribution of objects to enhance the aesthetic appearance of designs [24] (although further research is required in these areas). Moreover, whilst the emphasis of the research was focused within a creative domain, the findings may also present broader accessibility opportunities in other mainstream applications (e.g. word processors, presentation software, etc.) where snapping features are typically available (i.e. in terms of manipulating the layout of images and text).

One limitation of the work is the accuracy of speech recognition which is a known challenge within the field and can influence the usability of systems [34, 35]. Whilst the recognition accuracy was high (approximately 95%), there were still occasions where users had to repeat commands to perform different actions. For instance, similar sounding commands such as "Q" and "U" (which refer to snap point labels) were occasionally interpreted incorrectly. Related voice control systems (used by people with disabilities) such as Talon [36] use a phonetically diverse list of words for typing characters which contain a smaller number of syllables (as compared to NATO phonetic alphabets) [37]. It will therefore be important in future work to explore a set of commands that are efficient to pronounce and phonetically diverse (i.e. in terms of containing fewer syllables [42], being easy to recall). The study also only involved native English speakers as participants, so it will be important to evaluate the system performance using different languages in future studies. Another limitation is related to the resize tasks – whilst we covered a wide range of common sizes for interface elements (informed through analysis of visual elements in mainstream applications), it will also be important to cover a wider range of scenarios. For instance, the smallest object size in this study was 35x35 pixels, but it will be useful to explore attempting to resize objects to much smaller sizes to investigate any impact on the efficacy of the approaches developed. Furthermore, it will be important to explore the potential of voice snapping in terms of resizing a wider variety of interface elements (e.g. custom shapes and text) to examine whether this presents any unique interaction challenges that require further consideration.

Whilst the results highlighted an overall benefit for AutoSnap, there is still the possibility that UserSnap can be a more efficient and effective approach in some scenarios. This was highlighted through feedback from participants who felt that UserSnap was more appropriate when looking to adjust the corner transformation handles on objects (to avoid potential undesired automatic snapping to both vertical and horizontal snap points). This will also likely be the case in scenarios where multiple snap points are located in close proximity to each other – automatic snapping here may well lead to user frustration as it increases the likelihood that objects will snap to the incorrect location. Conversely, UserSnap may present benefits here as it would provide users with full control over which snap point they wish to target. A hybrid approach utilizing some degree of user control and automation may likely be optimal in certain scenarios, although additional research is required to empirically investigate this further and understand the nuances around object snapping via speech control. Future work also needs to explore potential adaptations to the voice commands used – for instance, participants used directional commands on occasions (e.g. "left big"), as well as chaining different commands together (e.g. "4 big 10"). The current system did not support these types of commands, hence further research around these areas could inform and enhance the usability of the existing approaches developed. Moreover, related research has previously explored the potential of vocal commands to augment the workflow of non-disabled professional designers alongside traditional input devices (i.e. a mouse, keyboard, stylus) [12–15]. AutoSnap and UserSnap may therefore also have wider potential to enhance the creative flow of non-disabled designers, although further work is required to confirm whether this may present interaction benefits.

7 Conclusion

We developed and evaluated three different speech controlled interaction techniques (NoSnap, UserSnap, and AutoSnap) for supporting people with physical impairments in resizing graphical objects located on a digital canvas. Results highlighted that participants found the AutoSnap approach to be more efficient, accurate, and usable than the other two approaches. Subjective feedback also confirmed that the AutoSnap approach was perceived positively and presented benefits over NoSnap and UserSnap. This work therefore demonstrates the benefit of tailoring common snapping features integrated within mainstream applications for voice interaction to support the development of more inclusive design environments.

REFERENCES

[1] Chris Creed. 2018. Assistive technology for disabled visual artists: exploring the impact of digital technologies on artistic practice. *Disability and Society.* 1103–1119. DOI:https://doi.org/10.1080/09687599.2018.1469400

[2] Chris Creed, Russell Beale, and Paula Dower. 2014. Digital tools for physically impaired visual artists. In *Proceedings of the 16th international ACM SIGACCESS conference on Computers & accessibility*. 253–254. DOI:https://doi.org/10.1145/2661334.2661386

[3] Susumu Harada, Jacob O. Wobbrock, Jonathan Malkin, Jeff A. Bilmes, and James A. Landay. 2009. Longitudinal study of people learning to use continuous voice-based cursor control. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems.* 347–356. DOI:https://doi.org/10.1145/1518701.1518757

[4] Susumu Harada, T Scott Saponas, and James A Landay. 2007. VoicePen: Augmenting pen input with simultaneous non-linguistic vocalization. In *Proceedings of the 9th International Conference on Multimodal Interfaces, ICMI'07*. 178–185. DOI:https://doi.org/10.1145/1322192.1322225

[5] Susumu Harada, Jacob O. Wobbrock, and James A. Landay. 2007. VoiceDraw: A hands-free voice-driven drawing application for people with motor impairments. In *ASSETS'07: Proceedings of the Ninth International ACM SIGACCESS Conference on Computers and Accessibility*. 27–34. DOI:https://doi.org/10.1145/1296843.1296850

[6] Perera, D., Eales, R.J. and Blashki, K., 2009. Supporting the creative drive: investigating paralinguistic voice as a mode of interaction for artists with upper limb disabilities. *Universal Access in the Information Society*, *8*(2) .77-88.

[7] Patrick Baudisch, Edward Cutrell, Ken Hinckley, and Adam Eversole. 2005. Snap-and-go: helping users align objects without the modality of traditional snapping. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 301–310.

[8] Seongkook Heo, Yong-Ki Lee, Jiho Yeom, and Geehyuk Lee. 2012. Design of a shape dependent snapping algorithm. In *CHI'12 Extended Abstracts on Human Factors in Computing Systems*. 2207-2212.

[9] Marianela C. Felice, Nolwenn Maudet, Wendy E. Mackay, and Michel Beaudouin-Lafon. 2016. Beyond snapping: Persistent, tweakable alignment and distribution with StickyLines. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. 133–144.

[10] Chris Creed, Ian Williams, and Maite Frutos-Pascual. 2020. Multimodal Gaze Interaction for Creative Design. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–13. DOI:https://doi.org/10.1145/3313831.3376196

[11] Jan Van der Kamp and Veronica Sundstedt. 2011. Gaze and voice controlled drawing. In *Proceedings of the 1st conference on novel gaze-controlled applications.* 1–8. DOI:https://doi.org/10.1145/1983302.1983311

[12] Gierad Laput, Mira Dontcheva, Gregg Wilensky, Walter Chang, Aseem Agarwala, Jason Linder, and Eytan Adar. 2013. PixelTone: A multimodal interface for image editing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2185–2194. DOI:https://doi.org/10.1145/2470654.2481301

[13] Arjun Srinivasan, Mira Dontcheva, Eytan Adar, and Seth Walker. 2019. Discovering natural language commands in multimodal interfaces. *In Proceedings of the 24th International Conference on Intelligent User Interfaces*. 661–672. DOI:https://doi.org/10.1145/3301275.3302292

[14] Jana Sedivy and Hilary Johnson. 1999. Supporting creative work tasks: the potential of multimodal tools to support sketching. In *Proceedings of the 3rd conference on Creativity and Cognition*. 42–49.

[15] Yea Seul Kim, Mira Dontcheva, Eytan Adar, and Jessica Hullman. 2019. Vocal shortcuts for creative experts. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–14. DOI:https://doi.org/10.1145/3290605.3300562

[16] Designing Out Loud: Announcing Support for macOS Voice Control in Adobe XD. 2020. Retrieved January 21, 2022 from https://blog.adobe.com/en/2020/02/11/announcing-mac-os-voice-control-adobe-xd.html?scid=fac788f5-fe6f-4be4-a960-871ac58b5f30&mv=social&mv2=owned_social

[17] Eric A. Bier and Maureen C. Stone. 1986. Snap-dragging. *ACM SIGGRAPH Computer Graphics*, *20*(4). 233–240.

[18] Toshiyuki Masui. 2001. September. HyperSnapping. In *Proceedings IEEE Symposia on Human-Centric Computing Languages and Environments (Cat. No. 01TH8587)*. 188–194.

[19] Shari Trewin, Simeon Keates, and Karyn Moffatt. 2006. October. Developing steady clicks: a method of cursor assistance for people with motor impairments. In *Proceedings of the 8th International ACM SIGACCESS Conference on Computers and Accessibility*. 26–33.

[20] Jennifer Fernquist, Garth Shoemaker, and Kellogg S. Booth. 2011. September. "Oh Snap"– Helping Users Align Digital Objects on Touch Interfaces. In *IFIP Conference on Human-Computer Interaction*. Springer, Berlin, Heidelberg. 338–355.

[21] Anastasia Schaadhardt, Alexis Hiniker, and Jacob O. Wobbrock. 2021. Understanding Blind Screen-Reader Users' Experiences of Digital Artboards. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems.* 1–19.

[22] Mathias Frisch, Sebastian Kleinau, Ricardo Langner, and Raimund Dachselt. 2011. Grids & guides: multi-touch layout and alignment tools. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1615–1618.
[23] Roope Raisamo, and Kari-Jouko Räihä. 1996. A new direct manipulation technique for aligning objects in drawing programs. In *Proceedings of the 9th annual ACM symposium on User interface software and technology*. 157–164.
[24] Pengfei Xu, Hongbo Fu, Takeo Igarashi, Chiew-Lan Tai. 2014. Global beautification of layouts with interactive ambiguity resolution. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*. 243–252.
[25] Using the Web Speech API - Web APIs | MDN. Retrieved January 21, 2022 from https://developer.mozilla.org/enUS/docs/Web/API/Web_Speech_API/Using_the_Web_Speech_API
[26] Zoom. 2011. Video Conferencing, Web conferencing, Webinars, Screen Sharing – Zoom. Retrieved January 21, 2022 from https://www.zoom.us/
[27] Aaron Bangor, Philip Kortum, and James Miller. 2009. Determining what individual SUS scores mean: adding an adjective rating scale. Journal of usability studies. 114–123.
[28] Andrew Sears, Min Lin, and Azfar S. Karimullah. 2002. Speech-based cursor control: understanding the effects of target size, cursor speed, and command selection. Universal Access in the Information Society. 30–43. DOI:https://doi.org/10.1007/s10209-002-0034-6
[29] Samuel S. Shapiro, and Martin B. Wilk. 1965. An Analysis of Variance Test for Normality (Complete Samples). Biometrika. 591–611. DOI:https://doi.org/10.2307/2333709
[30] Farkhandah Aziz, Chris Creed, Maite Frutos-Pascual, and Ian Williams. 2021. Inclusive Voice Interaction Techniques for Creative Object Positioning. In *Proceedings of the 2021 International Conference on Multimodal Interaction*. 461–469.
[31] Mayumi Hiyoshi and Hideo Shimazu. 1994. Drawing pictures with natural language and direct manipulation. In COLING 1994 Volumn 2: The 15th International Conference on Computational Linguistics. DOI:https://doi.org/10.3115/991250.991262
[32] Per-Erik Danielsson. 1980. Euclidean distance mapping. *Computer Graphics and image processing*, *14*(3). 227–248.
[33] Donald W. Zimmerman and Bruno D. Zumbo. 1993. Relative power of the Wilcoxon test, the Friedman test, and Repeated-measures ANOVA on ranks. The Journal of Experimental Education. 75–86. DOI:https://doi.org/10.1080/00220973.1993.9943832
[34] Takuya Nishimoto, Nobutoshi Shida, Tetsunori Koayashi, and Katsuhiko Shirai. 1995. Improving human interface drawing tool using speech, mouse and key-board. In *Proceedings 4th IEEE International Workshop on Robot and Human Communication*. 107–112.
[35] Mohammad M. Alsuraihi, and Dimitris I. Rigas. 2007. How effective is it to design by voice?. In *Proceedings of HCI 2007 The 21st British HCI Group Annual Conference University of Lancaster, UK 21*. 1–4.
[36] Talon. 2022. Talonvoice. *Talon*. Retrieved January 21, 2022 from https://talonvoice.com.
[37] Saumya Solanki, Gautam Krishnan, Varshini Sampath, and Jason Polakis. 2017. In (cyber) space bots can hear you speak: Breaking audio captchas using ots speech recognition. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. 69-80.
[38] L. Dai, R. Goldman, A. Sears, and J. Lozier. 2005. Speech-based cursor control using grids: modelling performance and comparisons with other solutions. *Behaviour & Information Technology*, *24*(3), 219–230.
[39] Carlos Duarte and Joana Neca. 2011. Evaluation of Gestural Interaction with and without Voice Commands. In Proceedings of IHCI 2011-IADIS conference on Interfaces and Human Computer Interaction. Retrieved January 21, 2022 from https://www.researchgate.net/publication/256007049
[40] Adam S. Williams, and Francisco R. Ortega. 2020. Understanding gesture and speech multimodal interactions for manipulation tasks in augmented reality using unconstrained elicitation. In *Proceedings of the ACM on Human-Computer Interaction*, *4*(ISS). 1–21.
[41] Adam S. Williams, Jason Garcia, and Francisco Ortega. 2020. Understanding Multimodal User Gesture and Speech Behavior for Object Manipulation in Augmented Reality Using Elicitation. *IEEE Transactions on Visualization and Computer Graphics*, *26*(12). 3479–3489.
[42] Monika Elepfandt, and Martin Grund. 2012. Move it there, or not? The design of voice commands for gaze with speech. In *Proceedings of the 4th workshop on eye gaze in intelligent human machine interaction*. 1–3.
[43] Minkyung Lee, and Mark Billinghurst .2008. A wizard of oz study for an AR multimodal interface. In *Proceedings of the 10th international conference on Multimodal interfaces*. 249–256.
[44] Adobe. 1999. Photoshop apps - desktop, mobile, and tablet | Photoshop.com. Retrieved January 21, 2022 from https://www.adobe.com/products/photoshop.html
[45] Adobe Inc. 2022. Adobe Illustrator CS6: Industry-leading vector graphics software. Retrieved January 21, 2022 from https://www.adobe.com/uk/products/illustrator.html
[46] Adobe Inc. 2022. Adobe XD | Fast & Powerful UI/UX Design & Collaboration Tool. Retrieved Januray 21, 2022 from https://www.adobe.com/uk/products/xd.html
[47] Figma. 2022. Figma: The collaborative interface design tool. Retrieved January 21, 2022 from https://www.figma.com
[48] Nuance Communications. 2022. *Dragon Speech Recognition - Get More Done by Voice | Nuance*. Retrieved January 21, 2022 from https://www.nuance.com/dragon.html
[49] M. Dellisanti, M. Fiorentino, G. Monno, and A.E. Uva. 2008. Enhanced 3D object snap for CAD modelling on large stereo displays. *International journal of computer applications in technology*, *33*(1). 54–62. DOI:https://doi.org/10.1504/IJCAT.2008.021885
[50] Mirjam Augstein, Thomas Neumayr, Thomas Burger, Josef Altmann, Werner Kurschl, and Stephan Vrecer. 2018. Haptic and Touchless User Input Methods for Simple 3D Interaction Tasks: Interaction Performance and User Experience for People with and Without Impairments. In *International Joint Conference on Computer Vision, Imaging and Computer Graphics*. Springer, Cham. 51–81
[51] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. Qualitative Research in Psychology. 77–101. DOI:https://doi.org/10.1191/1478088706qp063oa