Squidgets: Sketch-based Widget Design for Scene Manipulation Supplemental Material

A Curve Matching Algorithm

A.1 Implementation Details

NPR Outlines: Non-photorealistic rendering (NPR), particularly toon rendering, employs stylized outlines to emphasize shape features and enhance visual clarity. Traditional NPR methods extract perceptually significant curves such as silhouettes, occluding contours, and suggestive contours from 3D geometry, which are subsequently stylized into expressive strokes, mimicking hand-drawn illustrations [2, 5]. Silhouettes and suggestive contours are typically computed using differential geometry to identify regions where surface orientation undergoes rapid changes, capturing key shape features. In our system, we use Maya's toon outline command `pfxToon` to extract our abstract curves from scene objects, which are further post-processed to establish vertex correspondences between a stroke and a curve.

Corner matching: Methods designed for polylines and smooth curves primarily analyze local geometric features, notably local curvature and tangent direction. Techniques that utilize curvature extrema or changes in tangent direction robustly detect stable corners along curves. To match detected corners between shapes, local descriptors such as Shape Contexts [1, 4] are employed, capturing spatial configurations in descriptor histograms to robustly establish correspondences.

Neighborhood search: In our context, neighborhood search methods enhance the matching of user-drawn curves with existing curves within a scene. Initially, a local search identifies candidate correspondences based on descriptor similarity as described in Section 4.1 of the paper. Subsequently, these matches are refined using neighborhood consistency checks to ensure geometric coherence. In our implementation, we naively use our curve descriptor similarity to match a user stroke to squidget curves on screen. This can be further optimized to match only the k closest curves since user stroke proximity conveys higher correspondence intent.

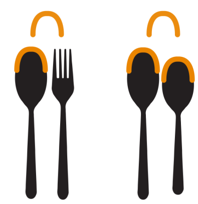
A.2 Use Cases

As described in Section 4.1 of the paper, a general similarity metric for curve matching is an ill-posed problem, depending on its use case and assumptions on the geometric properties of the curves. For most interactive graphics applications, curve matching is based on *corner*, *shape*, and *spatial* similarity [4].

Perceptually, we tend to align curves at corresponding sharp *corners* with matching smooth curve segments *shapes*. Depending on where the matching curves are drawn, *spatial* transforms may be needed to better align corresponding shape segments. Our curve matching algorithm for two curves P and Q in Section 4.1 is a two-step algorithm: 1) the curves are processed for internal corners, segmenting them into piecewise smooth curve segments, and a correspondence between corners of the two curves provides sparse anchors for curve matching; and 2) a matching energy is computed for each pair of corresponding curve segments as a rigid alignment *spatial* energy, and a distance between (i) the corresponding curve points *shape* energy and (ii) the energies for all corresponding curve segments summed.

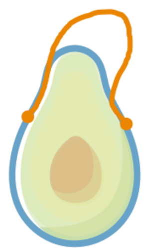
The weight parameters for the energies of each matching step can be defined and aspects of the algorithm simplified based on various use case scenarios. Below we list a few common curve interaction scenarios, including those used in our paper:

1-stroke squidget interaction: Using a single stroke S to specify both a curve selection P and its desired manipulated result S in general is likely to lead to ambiguities in curve selection, especially if the collection of curves being matched $\mathscr{C}$ contains one or more curves Q that better match S than the desired selection P. In the presence of this ambiguity, the onus is thus on the user to use this technique when the curves in $\mathscr{C}$ are very different from a desired manipulation S, or to use the technique for small changes to P, but few assumptions can be made *a-priori* on the relative importance of corner, spatial, or shape weighting.



2-stroke squidget interaction: The sole purpose of the first stroke S_s in 2-stroke interaction (i.e. select→manipulate) is to select a curve P from the curve collection $\mathscr{C}$. The expectation is such that a user will trace over the desired curve P as closely as possible. For this reason, articulated rigid matching of corners in unnecessary and the two strokes S_s and P can simply be treated as smooth strokes bypassing the corner correspondence algorithm, treating the closest pair of points on the two curves as the anchors for curve correspondence. No allowance for spatial alignment is necessary, and the curve segment rigid alignment transformation can be skipped altogether, or the alignment w_a can be set high. In essence, curve matching here degenerates to finding a closest pair of points on the two curves and then computing an average arc-length based distance between corresponding points on S_s and P. Now given S_s and S_m, the corner matching step of the curve matching algorithm can be applied to account for feature correspondence between the two curves, so that features on S_s parametrically transform to corresponding ones on S_m beyond just the shape of the curve.

Smooth stroke and corner gestures: Often gestures [6] are designed as a single smooth stroke for drawing efficiency. In such case, the corner matching step degenerates to finding the closest pair of points on the stroke to a curve being matched as an anchor prior to curve segment matching. Smooth curve segments meeting at sharp corners also provide a perceptually powerful feature for designing gestures. For gestures dominated by corner features, the corner matching energy can suffice to distinguish the desired corner configuration from a collection of curves, making further curve segment matching unnecessary.



Stroke-based local editing: Stroke-based shape deformations (Figure 12) are often indicated by localized editing of a curve. In such cases, the start and end of the edit stroke largely coincide with the curve being edited. Here, a weighted sum of curve segment matching energies can be used, with the end segments of the stroke have a higher weight than the interior. Further, rigid alignment for corner or curve segment matching should not be necessary.

Stroke-based translation or rotation: In a scenario where stroke-based transformation is constrained, aspects of rigid alignment for either or both the corner and curve segment matching steps can be reweighted (or skipped).

Corner-based selection: Smooth curve segments meeting at sharp corners provide perceptually powerful features for shape matching. We observed that users will often draw a stroke with a single corner, to suggest a similar corresponding corner on a curve in a collection $\mathscr{C}$.

B Usability Study: Design

We conducted a within-subject controlled experiment to assess the usability of different types of squidgets compared to a baseline, on typical scene manipulation tasks. Twelve (12) participants were recruited via announcements on our institutional social media channels and word of mouth.

B.1 Task Design

Our goal with the study was to gather initial understanding of the usability of the interaction paradigm which squidgets enable, and the characteristics of the scenes which may influence task performance and user experience. While the squidgets framework (and our implementation) support 3D scenes, we created tasks in a 2D view to remove navigation complexity of 3D environments, and spatial reasoning in 3D.

Participants were told: "In this study, you are commissioned to help design a summer drinks poster by moving fruit stickers onto the poster or changing fruit sticker shapes on the poster. To do so, you will use different techniques".

One task was to perform a **translation** of one object to a target location. Figure 1 shows an overview of the task components.

The object to move included a visual indicator within it, which participants were asked to align with the target indicator on the poster. We used a visual indicator as opposed to a "ghost image" of the target location, to avoid participants sketching over the target fruit. The visual indicator makes it somewhat hard for participants to precisely reconstruct where the object outline would be, thus potentially penalizing the squidget techniques in our study, compared to more realistic scenarios (e.g. sketch in relation to other objects present in the scene, such that objects align, or share a border, etc...).

Because our sketch tool must disambiguate selection amongst (possibly) other surrounding objects, we place 3 distractor images on top the poster in different locations for each task that the tool could mistakenly select.

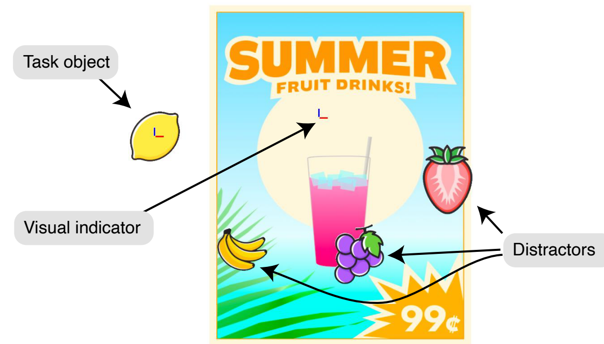


Figure 1: Translate Task Components.

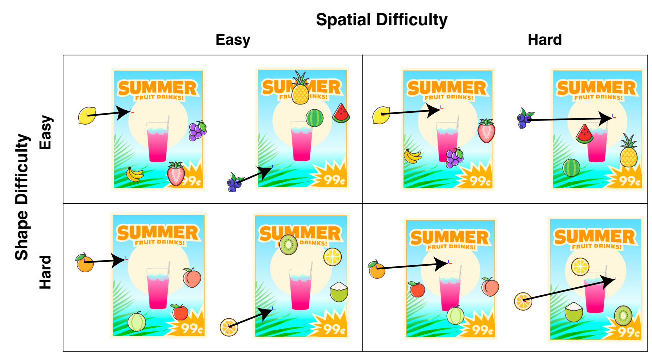


Figure 2: Translate Task: For each combination of `SpatialDifficulty` × `ShapeDifficulty`, participants completed two tasks.

The distractors made the task more difficult for squidgets, with regard to two factors. First, the selection of the object to move is potentially made more difficult, when other distractors are within the vicinity of the task object and target location. `SpatialDifficulty` is how far each distractor image is from the target axis location. We manipulate this variables according to two levels: Easy spatial difficulty places all distractors further away from the target location than the main image and Hard spatial difficulty places all distractors closer from the target location than the main image. See Figure 2.

The tasks could also be more difficult when the task object has similar visual features as other surrounding objects, making it potentially challenging for squidgets to disambiguate which object the user intended to manipulate. `ShapeDifficulty` sets how visually similar in shape the distractor image is to the task image. We set two levels of difficulty: Easy Shape difficulty where all 3 images are of visually different shape, and Hard Shape difficulty where all 3 images are of visually similar shape. Visual similarity was subjectively decided. See Figure 2.

For each shape difficulty we created set A and set B of fruit images, as showed in Figure 2.

The other task was to perform a **deformation** of one object to match a target shape. Figure 3 shows an overview of the task components.

For this task, we were mostly interested in whether participants were able to deform the shape by sketching the envisioned result,

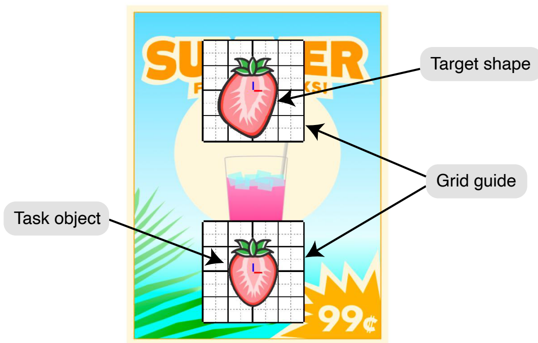


Figure 3: Deform Task Components.

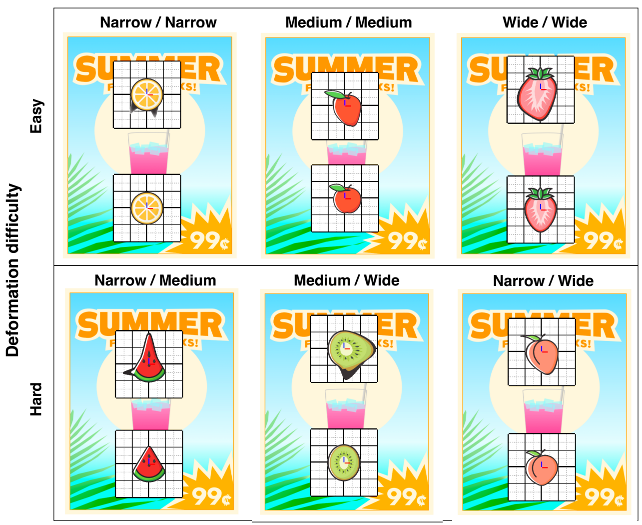


Figure 4: Deform Task: For each pair of {narrow, medium, wide} deformation type, we generated a task. Tasks where both deformation were of the same type are considered "easy" with respect to `DeformDifficulty`, and the others are considered "hard".

rather than being interested in disambiguating the selection from other distractor objects. Hence, we only included a task object, which participants had to deform using squidgets to match a target shape. We introduced a reference grid underneath the objects, because we found in our pilot sessions that matching perfectly the shape was difficult (for all techniques) in the absence of such a reference to guide precise deformations. We chose this task design over other options such as a ghost target shape which would be overlaid on top of the task shape, as this would unfairly make squidget techniques easier, since participants could simply trace over.

Like the translate tasks, we manipulated task difficulty for the deform tasks. For squidgets, the difficulty lies in how precisely participants can draw the intended target shape. We speculate that the more changes, and the more different the changes, the finer motor skills are required. For instance, adding a small spike to a fruit is less complex to realize than adding two small spikes, which is also less complex to realize than adding a small spike and squeezing or stretching the whole fruit too. For the baseline, this also requires to change the parameters of the deformation tool, i.e. the "Vertex Falloff Radius" for the Mesh Deformation tool.

We designed the deform tasks such that the target shape would be the result of applying to the task fruit two separate deformations. Each deformation could be either narrow (i.e. 0.5), medium (i.e. 1.5) or wide (i.e. 2.5). We created the target shape using Maya's vertex deformation tool, varying the combinations of narrow, medium, and wide deformations. Figure 4 shows all 6 tasks that we generated.

B.2 Implementation

The study was created in Maya 2024 because the tool was originally developed in Maya 2024. We recognize that Maya 2024 is an industry-level software that many users do not have experience with; however, many people are familiar with simple artistic software (Microsoft Paint) or drawing tools (pen tablets). To present a more familiar graphical layout, we presented a stripped down layout of Maya were users were shown only the scene window and basic tools akin to traditional vector graphics software.

B.3 Study Setup

The study ran on a computer laptop plugged to Wacom Cintiq 24HD screen tablet (touch and pen input); and was implemented with the Maya software, with most features stripped out from the view to best approximate a generic, traditional vector graphics editor. We chose to conduct the study using a pen-interactive display over mouse or drawing tablet because pen-interactive displays provide a better feel of direct manipulation and control over other mediums for drawing tasks. While we do recognize that pen-interactive displays are not particularly accessible, applications with drawing interactions through touch screens or tablet pens is widely common.

The squidgets framework is general and its application space vast. The goal of this controlled experiment was to evaluate whether people understand the concept of squidgets, and the usability of using sketch input for manipulating scene parameters. We opted for simple tasks created in a 2D view to remove navigation complexity of 3D environments. We explore squidgets use in 3D scenes in a more ecologically-valid context in Section 7.2 of the paper. The experiment was instrumented to log interaction events, including study progression (e.g., start, end, phase transitions), device input (e.g., pen inputs), and squidget-specific actions, commands, and metrics. Responses to questionnaires were collected via digital forms. The sessions lasted approximately 45 minutes. Participants were compensated CAD$30 in appreciation for their time. The study was approved by the boards of ethics at our institution.

B.4 Experimental Design

Participants were asked to complete two sets of common scene manipulation `Tasks`:

- *translate*: Translate a graphical object to a target location.
- *deform*: Deform a graphical object to match a target.

using different `Techniques`:

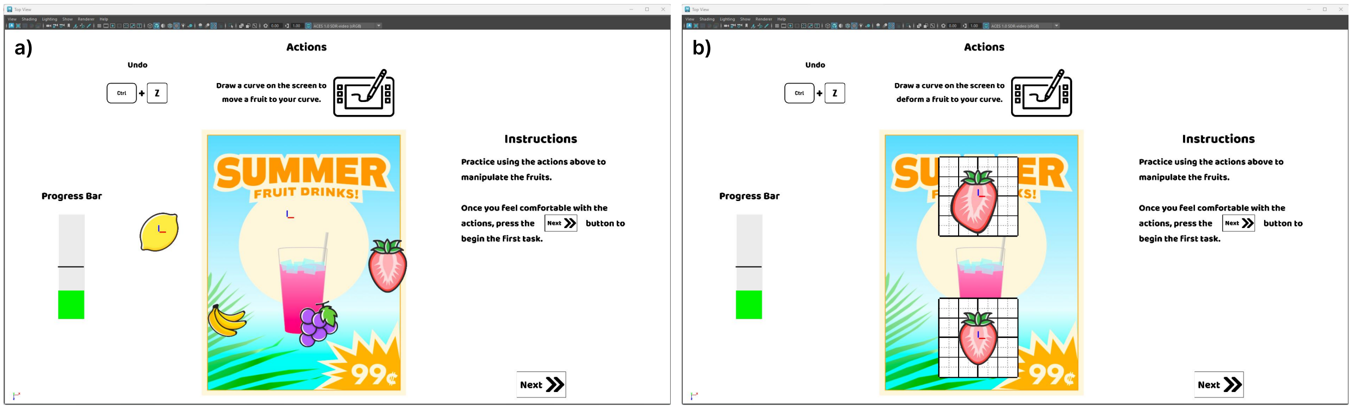


Figure 5: Study interface for translation tasks (a) and deformation tasks (b).

- *baseline*: Traditional graphical manipulation tools, as found in vector graphics software.
- *1-stroke*: A squidget where both selection and manipulation are performed using a single input stroke (i.e. $S_S = S_M$).
- *select→drag*: A squidget where one input stroke is used for selection (i.e. S_S), upon which the object sticks to the cursor and can be further dragged around (only used for the translate Task)
- *select→manipulate*: An initial stroke is used for selection (S_S). Subsequent drawn strokes perform manipulations to the selected object (S_M).

We provided participants with feedback on how close/far they were to completing a given task. For the translation task, we used the Euclidean distance between the axes of the task image and the target location axes. For the deformation task, we used the Intersection over Union (IoU) of the two image shapes. Figure 5 shows screenshots of the study interface for both tasks.

For the translation task, each participant completed a total of: 4 `Techniques` × 2 `SpatialDifficulty` × 2 `ShapeDifficulty` × 2 repetitions = 32 trials.

For the deformation task, the *select→drag* technique is irrelevant and was thus excluded. Participants completed a total of: 3 `Techniques` × 2 `DeformDifficulty` × 3 repetitions = 18 trials.

B.5 Data Collection

For each trial, we collected the task completion `Time` (i.e. the duration between the first and last interaction involving an object), the `Operation` count (i.e., click, click+drag, each counted as one operation), as well as the `Undo` count. We also administered the NASA Task Load Index (NASA-TLX) questionnaires after each technique block, and collected subjective rankings and open comments from participants in a free-form digital form.

C Usability Study: Results

We report the 95% confidence intervals (CI) on means to assess effects [3] along with qualitative feedback. All CIs are calculated using the studentized bootstrapping method.

C.1 Translation Task: Participants' Drawn Strokes

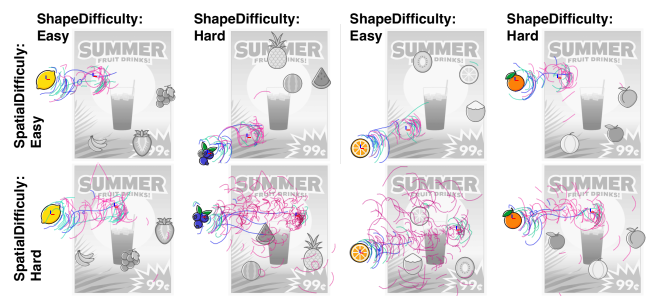


Figure 6: Move Task: Participants' sketch strokes for *1-stroke*, *select→drag*, and *select→manipulate*.

Figure 6 overlays all participants' sketch strokes for all three techniques. Overall, participants drew short arc-shaped strokes to perform the tasks. We observed that when distractors were spatially close to the target, participants tended to draw "incremental" simple strokes to move the object progressively closer to the target location, instead of attempting to fine-tune the stroke to incorporate distinguishable features for disambiguation, e.g. drawing a larger part of the fruit.

C.2 Translation Task: Completion Time

C.2.1 Completion time by Technique. Measurements per cell: 96.

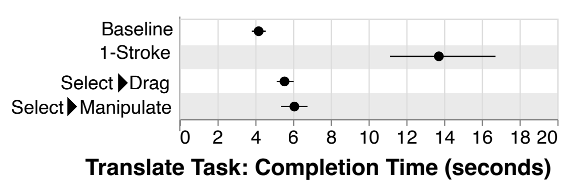


Figure 7: `Completion Time` for the `translation` task, for each `Technique`.

- *baseline* yielded the best completion time performance: 4.18s, CI [3.82, 4.56]. The *select→drag* and *select→manipulate* techniques closely followed, both having comparable completion times: 5.55s, CI [5.14, 6.04] and 6.07s, CI [5.37, 6.77] respectively. The *1-stroke* technique was the slowest, by far: 13.73s, CI [11.13, 16.72].

C.2.2 Effect of Spatial Difficulty on Completion time, by Technique. Measurements per cell: 48.

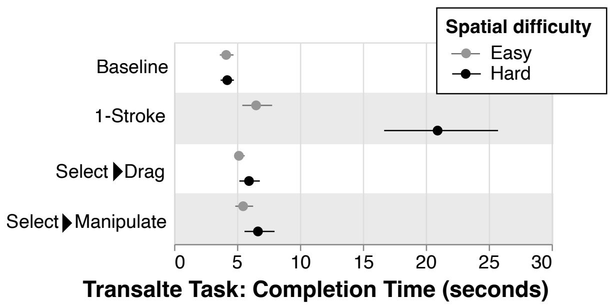


Figure 8: Effect of `SpatialDifficulty` on `Completion Time` for the `translation` task, for each `Technique`.

- *1-stroke* technique for tasks with **"hard" spatial difficulty** was much slower 20.94s, CI [16.69, 25.74], than for tasks with **"easy" spatial difficulty** 5.15s, CI [4.81, 5.61].
- For other techniques, the completion times were comparable between **"easy" spatial difficulty** and **"hard" spatial difficulty**, suggesting a negligible or no effect of spatial difficulty on the task for these techniques. For *baseline*, completion times were 4.14s, CI [3.63, 4.72] for easy spatial difficulty, and 4.22s [3.70, 4.75]. For the **select-drag** techniques, completion times were 5.14s, CI [4.81, 5.61] for easy spatial difficulty, and 5.95s [5.18, 6.81] for hard spatial difficulty. And for *select⟶manipulate*, completion times were 5.48s, CI [4.86, 6.28] for easy spatial difficulty, and 6.66s [5.59, 7.98] for hard spatial difficulty.

C.2.3 Effect of Shape Difficulty on Completion time, by Technique. Measurements per cell: 48.

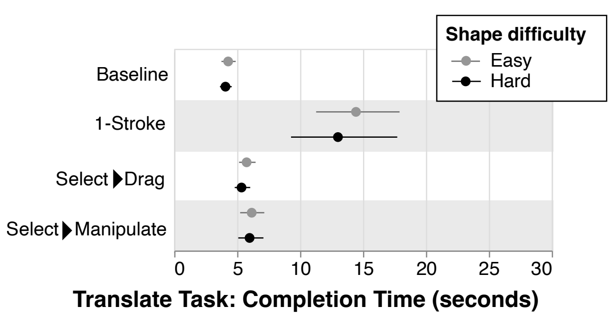


Figure 9: Effect of `ShapeDifficulty` on `Completion Time` for the `translation` task, for each `Technique`.

- There was strong evidence that **shape difficulty** did not impact completion times across techniques.
- For the *baseline*, times were comparable between easy shape difficulty 4.29s, CI [3.75, 4.89] and hard shape difficulty 4.07s, CI [3.63, 4.57]. Similarly, for the *1-stroke* technique, completion times were comparable between easy shape difficulty 14.44s, CI [11.28, 17.91] and hard shape difficulty 13.01s, CI [9.28, 17.72]. For the *select⟶drag* technique, completion times were also comparable between easy shape difficulty 5.75s, CI [5.15, 6.47] and hard shape difficulty 5.35s, CI [4.81, 6.04]. Finally shape difficulty did not have an effect on completion time for the *select⟶manipulate* technique, with comparable values for easy shape difficulty 6.16s, CI [5.24, 7.16] and hard shape difficulty 5.99s, CI [5.09, 7.09].

Takeaway insights:

1. Spatial difficulty has a strong effect on completion time for the *1-stroke* squidget technique, with a duration increase for tasks of "hard" spatial difficulty around three- to four-fold that of "easy" spatial difficulty or any other technique.
2. The traditional translate tool (i.e. *baseline*) was slightly faster than using squidgets overall.

C.3 Translation Task: Operations and Undos

We observe a similar notable trend as completion times for the other two dependent variables `Operation` count and `Undo` count, that is, the *1-stroke* squidget technique performed less well than other techniques.

C.3.1 Operation count by Technique. Measurements per cell: 96.

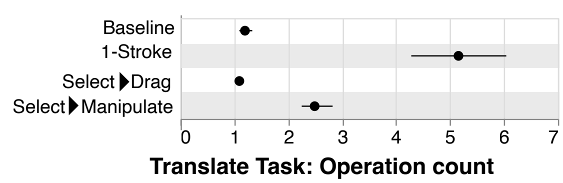


Figure 10: `Operation` count for the `translation` task, for each `Technique`.

- The *1-stroke* technique yielded the most number of operations performed 5.16, CI [4.28, 6.04]. This is already more than twice as many operations as were performed using the *select⟶manipulate* technique 2.49, CI [2.25, 2.82]. The *baseline* and *select⟶drag* techniques both yielded the least number of operations, with comparable counts of 1.20, CI [1.09, 1.33] and 1.09, CI [1.03, 1.17] respectively.

A break down per spatial difficulty (see paper) and shape difficulty (see paper) shows that spatial difficulty is the main factor affecting performance (easy: 2.15, CI [1.77, 2.63]; hard: 8.17, CI [6.96, 9.50]).

C.3.2 Undo count by Technique. Measurements per cell: 96.

- The *1-stroke* technique yielded the most number of undos performed 0.45, CI [0.27, 0.64]. The other techniques are all comparable, with very low values. *baseline*: 0, CI [0,0];

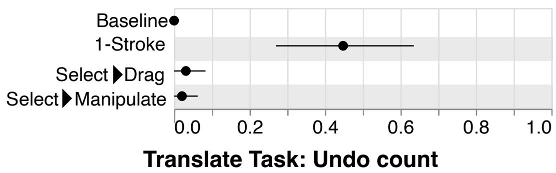


Figure 11: Undo count for the `translation` task, for each `Technique`.

select⟶drag: 0.031, CI [0, 0.08]; *select⟶manipulate*: 0.02, [0, 0.06].

Again, break down per spatial difficulty (see paper) and shape difficulty (see paper) shows that spatial difficulty is the main factor affecting performance (easy: 0.10, CI [0.02, 0.23]; hard: 0.79, CI [0.50, 1.10]).

Takeaway insights:

3. Spatial difficulty had a strong effect on operation count and undos, and thus on completion time for the *1-stroke* squidget technique.
4. We found strong evidence that the main contributing factor to decreased performance (time) for the *1-stroke* technique is the number of operations which participants performed, which significantly increased when the spatial difficulty was high, i.e. when distractors were close to the target location where the task object should be moved to.
5. While we find an effect on the undo count, we note that all techniques yielded less than one undo operation on average, which we note is generally low.

C.4 Deform Task: Participants Drawn Strokes

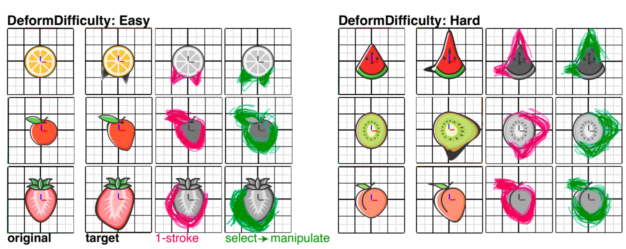


Figure 12: Deform Task: Participants' sketched strokes for *1-stroke*, and *select⟶manipulate*.

Figure 12 overlays all participants' sketch strokes using the *1-stroke* and the *select⟶manipulate* techniques for the deform task. For both techniques, we observed that when participants were not satisfied with their deformation, they would tend to undo the operation to start over again and repeat, until they succeeded in deforming the object. In contrast, with the *baseline*, participants proceeded with incremental adjustments using the distortion tool. We note that deformation with squidgets interactions relies entirely on the user's drawing skills, whereas operations performed with tools are more of a sculpting process, through stretch/push manipulations of the object's silhouette.

C.5 Deform Task: Completion Time

C.5.1 Completion time by Technique. Measurements per cell: 72.

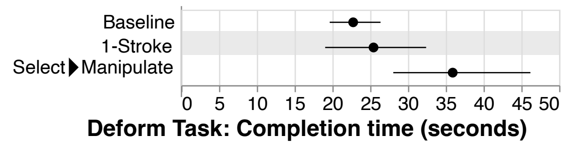


Figure 13: `Completion Time` for the `deformation` task, for each `Technique`.

- We found that *select⟶manipulate* takes significantly longer than the *baseline* technique for the deformation tasks overall, with values of 35.91s, CI [28.05, 46.20] for select-manipulate and 22.75s, CI [19.64, 26.37] for baseline. The *1-stroke* technique had completion times 25.44, CI [19.04, 32.40] that are comparable with baseline (i.e. the CIs mostly overlap), though there is weak evidence that it is slightly longer. There is also reasonable evidence that *1-stroke* performs better than *select⟶manipulate* (CIs overlap, but *select⟶manipulate* largely spans higher completion times.

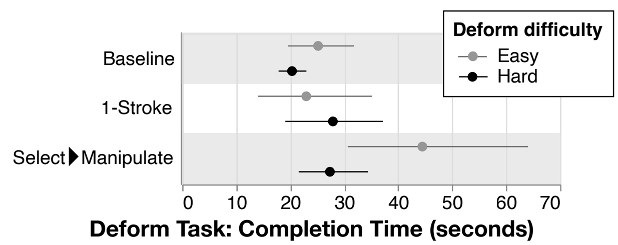


Figure 14: Effect of `Deform Difficulty` on `Completion Time` for the `translation` task, for each `Technique`.

- We find weak evidence that it takes less time to complete tasks whose difficulty is "hard", than that whose difficulty is "easy" within each technique. For *baseline*, hard tasks took an overall 20.33s, CI [17.84, 23.00], and easy tasks took an overall 25.16s, CI [19.57, 31.85]. For *1-stroke*, completion times were comparable between the hard (27.91s, CI [19.10, 37.16]) and the easy (22.97s, CI [14.01, 35.18]) tasks. For , there is reasonably strong evidence that hard tasks are about twice faster to complete (27.35s, CI [21.56, 34.37]) than the easy tasks (44.48s, CI [30.65, 64.04]).

Takeaway insights:

6. We found an inverse relation between task difficulty and performance measures, suggesting that our chosen measure is a poor proxy for gauging difficulty.
7. The types of deformations (i.e. same vs. different) have an effect on the *select⟶manipulate* squidget technique, with different deformations (i.e. "hard" tasks) being faster and requiring less operations to complete than tasks where deformations were the same (i.e. "easy" tasks).

C.6 Deform Task: Operation and Undo Counts

C.6.1 Operation count time by Technique. Measurements per cell: 72.

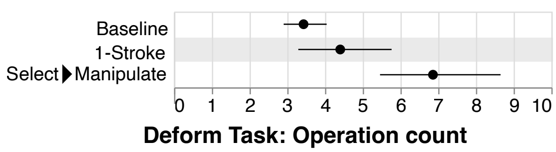


Figure 15: `Operation` count for the `translation` task, for each `Technique`.

- We found that *select⟶manipulate* takes significantly more operations than the *baseline* technique for the deformation tasks overall, with values of 6.86, CI [5.46, 8.65] for select-manipulate and 3.43, CI [2.90, 4.04] for baseline. The *1-stroke* technique lies in between with no significant differences with either other technique (i.e. CIs overlap): 4.40, CI [3.29, 5.76]. There is, however reasonable evidence that *1-stroke* performs better than *select⟶manipulate* (small overlap), and weak evidence that *1-stroke* performs less well than the *baseline* (large overlap, but 1-stroke CI extends to higher values).

C.6.2 Undo count time by Technique. Measurements per cell: 72.

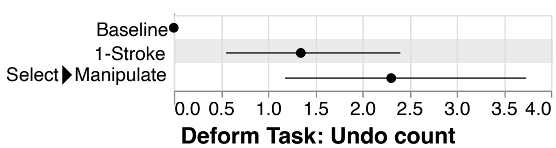


Figure 16: `Undo` count for the `translation` task, for each `Technique`.

- We found that *baseline* does not result in reverting operations: 0, CI [0,0].
- In contrast, both squidget techniques resulted in participants reverting operations before trying again. While there is no significant differences between the two squidget techniques, there is weak evidence that *1-stroke* resulted in less undos 1.35, CI [0.56, 2.40] than **stroke-manipulate** 2.30, CI [1.18, 3.74].

When breaking down per task difficulty (see paper), we observe that the types of deformation (i.e. same vs. different) have an effect on , both in terms of operation counts (different/hard: 5.39, CI [4.33, 6.56]; same/easy: 8.33, CI [5.72, 11.75]), and undo count (different/hard: 1.19, CI [0.53, 2.11] ; same/easy: 3.41, CI [1.22, 6.58]), just like for completion time.

Takeaway insights:

8. We find that task difficulty affects both the number of operations, and undo operations for the *select⟶manipulate* technique, with "hard" tasks (different deformations) performing better than the "easy" tasks (same deformations). Naturally, completion times follow the same trend.
9. For both squidget techniques, participants went through trial-and-error, with multiple operations and a non-negligible number of them being reversed (undo). This contrasts with the *baseline*, for which participants continued to deform further, without reverting operations.

C.7 Subjective Rankings

Here, we report on the participants' rankings when asked: *"For only the Move tasks, rank the methods from most preferred (1 Top) to least preferred (4 Bottom)."*(Figure 17) and *"For only the Deform tasks, rank the methods from most preferred (1 Top) to least preferred (3 Bottom)."* (Figure 18)

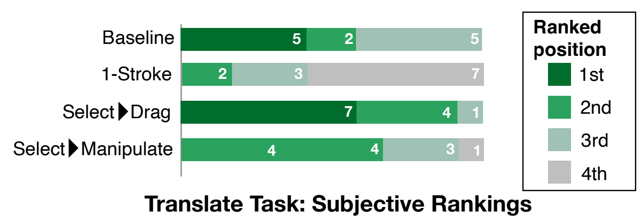


Figure 17: Participants' subjective ranking of the `Techniques`, to perform translation tasks.

Participants who ranked the baseline at the top felt that it gave them more control. P1 said that they *"prefered the default move tool because I felt like I had the most control over moving an object to where i wanted with most precision"*. Similarly, P5 indicated that *"default move had much freedom"* (P5).

Interestingly, participants who ranked the *select⟶drag* technique first listed similar reasons. P6 said *"stroke+drag method provides the user freedom to move the object with minimal actions than the other methods do."*. Similarly, P7 said *"Holding helped when moving the fruits around accurately"* and P2 *"stroke+drag was the fastest and the most intuitive, and also had very good control in where you want to move the object"*. P10 appreciated the real-time feedback: *"I like the stroke+drag one the most because you get to see real-time where the object is instead of hoping. It will be at the most correct place at the end, and it somehow just feels more natural than the default translation tool with the stroke action."*

We conclude that both approaches are complementary. With comparable performance, and perceived sense of control, our results suggest that both techniques could be useful to people, and choosing one over the other would mainly be motivated by personal preference.

In contrast, and in line with poor performance on the translation task, the task was found to be confusing: *"The [*select⟶manipulate *] technique was quite hard to understand in a short amount of time."* (P9), and P11 (who rated it second) indicated *"I would rate* select⟶manipulate *highly too, but it requires at least two strokes, where one stroke is selecting part of the object so it requires some attention to match the shape."*

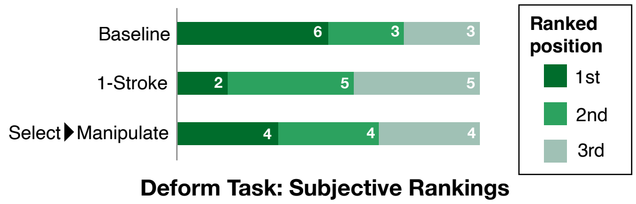


Figure 18: Participants' subjective ranking of the `Techniques`, to perform deform tasks.

The cons of using the baseline technique included the need to change parameters for the deformation: *"The [*baseline *] tool worked well but it required moving the pen around all the way to the left side [to change the parameter]"* (); and lack of control *" The default move tool seems to work well but it is very hard to gauge how wide "wide" is and sometimes guesses aren't very accurate. It is also a little difficult for me in a sense that it's hard to map a larger deformation of the image to just a single vertex movement in my head."* (P11).

Other participants felt the opposite was true. For instance P6 said *"The default move tool, by giving concrete options for the range of deformation, gives the user better sense to know how the object will be deformed than let the user define the area of the deformation."*.

Similarly, some participants preferred the *select→manipulate* tool because it matches better their expectations and thus gave them a high sense of control. P11: *"I like the [*select→manipulate *] tool the best because I feel like it provided the most amount of control - I can control the length of the curve that I want to deform as well as how much I want to deform it. "*. P8 said *" the pre select tool [*select→manipulate *] was more accurate in what area I wanted to deform."*. Whereas others found the opposite was true.

Here too, our results suggest that there is value in offering people the ability to perform the same tasks using different techniques. Different people will find different techniques best suit their needs.

Takeaway insights:

10. Overall, for the study tasks, participants consistently ranked the *baseline* highly.
11. For translation tasks, the *select→drag* was the most preferred, and the *1-stroke* was least preferred overall.
12. For deform tasks, the *baseline* was most preferred, followed by *select→manipulate*. *1-stroke* was least preferred, yet was ranked above one or both other techniques by 7 participants.
13. Different people feel differently about the techniques. We find opposite claims in terms of how much control different techniques afford. Bottom line is: each of the techniques has value, and feels best for different people. Which one to choose boils down to personal preference.

C.8 Subjective Workload (NASA-TLX)

Participants felt that *1-stroke* was more demanding than other techniques, as clearly reflected in the subjective NASA-TLX measures (see Figure 19-a). This technique was rated as more mentally and physically demanding, more effortful, more frustrating, and less performing than the other techniques for the move tasks. Most participants (7/12) rated this technique as their least preferred for these types of tasks, commenting that *"For [the 1-stroke] tool, the system kept misidentifying the object that I wanted to select, and also had to clear out the trajectory by moving other objects"* (P2). Contrarily, one of the two participants who rated this technique as second most preferred appreciated that *"being able to do things in one stroke was overall much smoother compared to selecting."* (P3).

D Study Material: Questionnaires

D.1 Task Survey

After each `Task` × `Technique` blocks, participants were asked to rate the technique using the NASA Task Load Index Questionnaire. Questions were as shown in Figure 20.

D.2 Post-Study Survey

Figure 21 shows the questionnaire which participants were asked to fill out, after completing the study. Results for subjective rankings are shown in Figure 17 and Figure 18.

References

[1] Serge Belongie, Jitendra Malik, and Jan Puzicha. 2002. Shape matching and object recognition using shape contexts. *IEEE transactions on pattern analysis and machine intelligence* 24, 4 (2002), 509–522.
[2] Doug DeCarlo, Adam Finkelstein, Szymon Rusinkiewicz, and Anthony Santella. 2003. Suggestive contours for conveying shape. *ACM Trans. Graph.* 22, 3 (July 2003), 848–855. https://doi.org/10.1145/882262.882354
[3] Pierre Dragicevic. 2016. Fair statistical communication in HCI. *Modern statistical methods for HCI* (2016), 291–330.
[4] Martin A. Fischler and Helen C. Wolf. 1994. Locating Perceptually Salient Points on Planar Curves. *IEEE Trans. Pattern Anal. Mach. Intell.* 16, 2 (1994), 113–129. https://doi.org/10.1109/34.273737
[5] Aaron Hertzmann. 1999. Introduction to 3d non-photorealistic rendering: Silhouettes and outlines. *Non-Photorealistic Rendering. SIGGRAPH* 99, 1 (1999).
[6] Jacob O. Wobbrock, Andrew D. Wilson, and Yang Li. 2007. Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes. In *Proc. of UIST '07*. 159–168. https://doi.org/10.1145/1294211.1294238

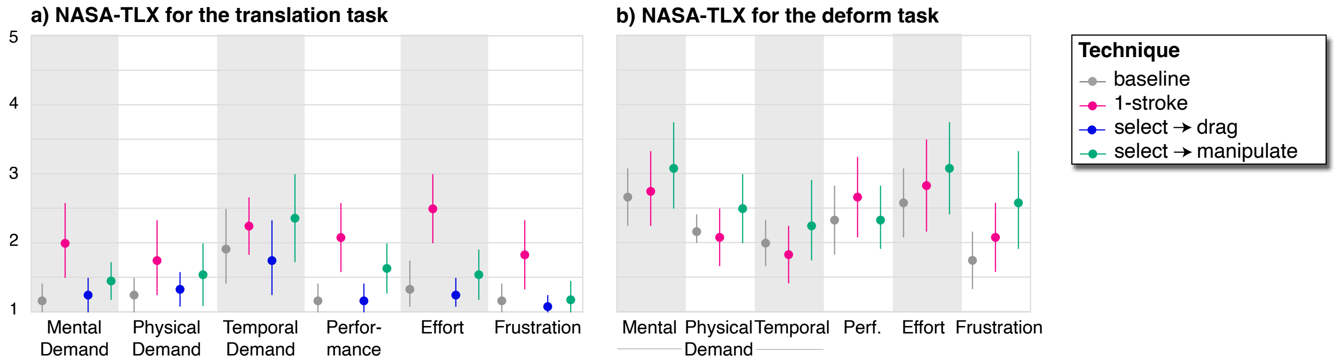
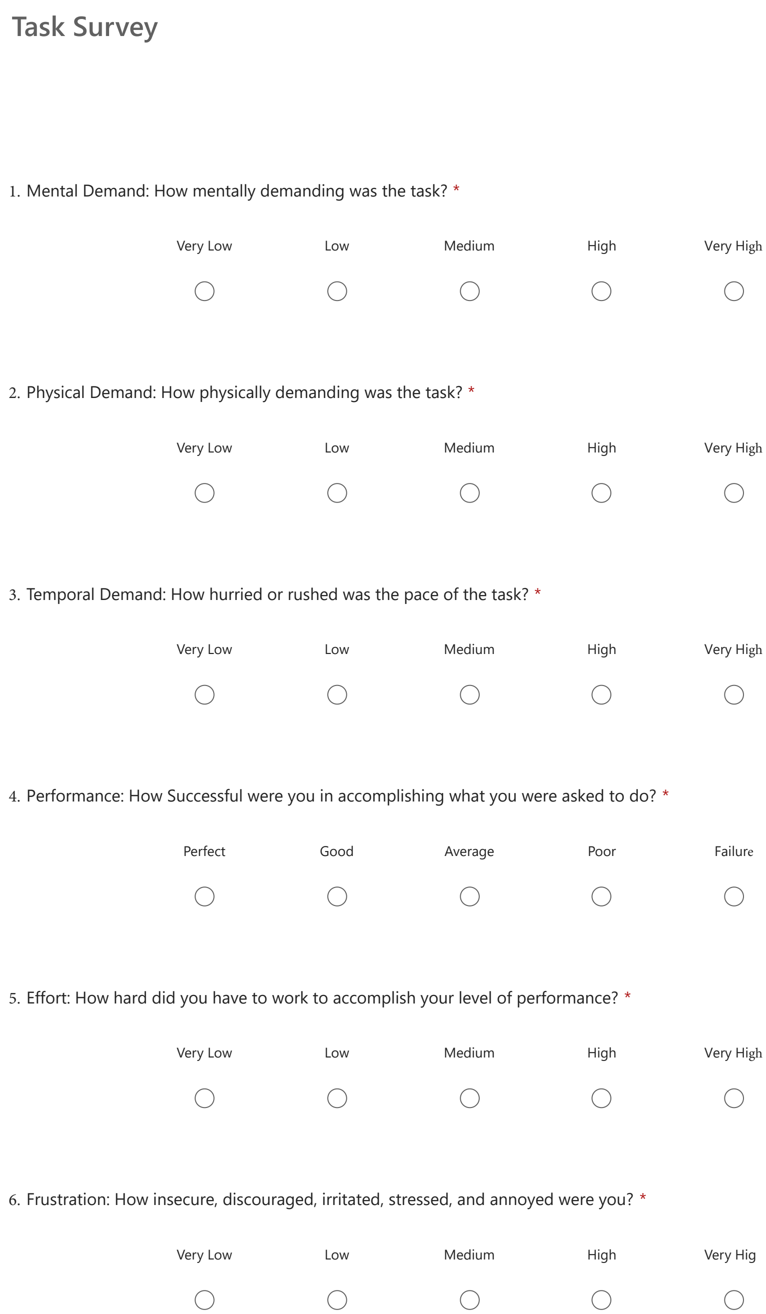


Figure 19: Effect of `Technique` on NASA-TLX measures (1=Very Low | Perfect; 5=Very High | Failure). The lower the value, the better.



Task Survey

1. Mental Demand: How mentally demanding was the task? *

Very Low	Low	Medium	High	Very High
○	○	○	○	○

2. Physical Demand: How physically demanding was the task? *

Very Low	Low	Medium	High	Very High
○	○	○	○	○

3. Temporal Demand: How hurried or rushed was the pace of the task? *

Very Low	Low	Medium	High	Very High
○	○	○	○	○

4. Performance: How Successful were you in accomplishing what you were asked to do? *

Perfect	Good	Average	Poor	Failure
○	○	○	○	○

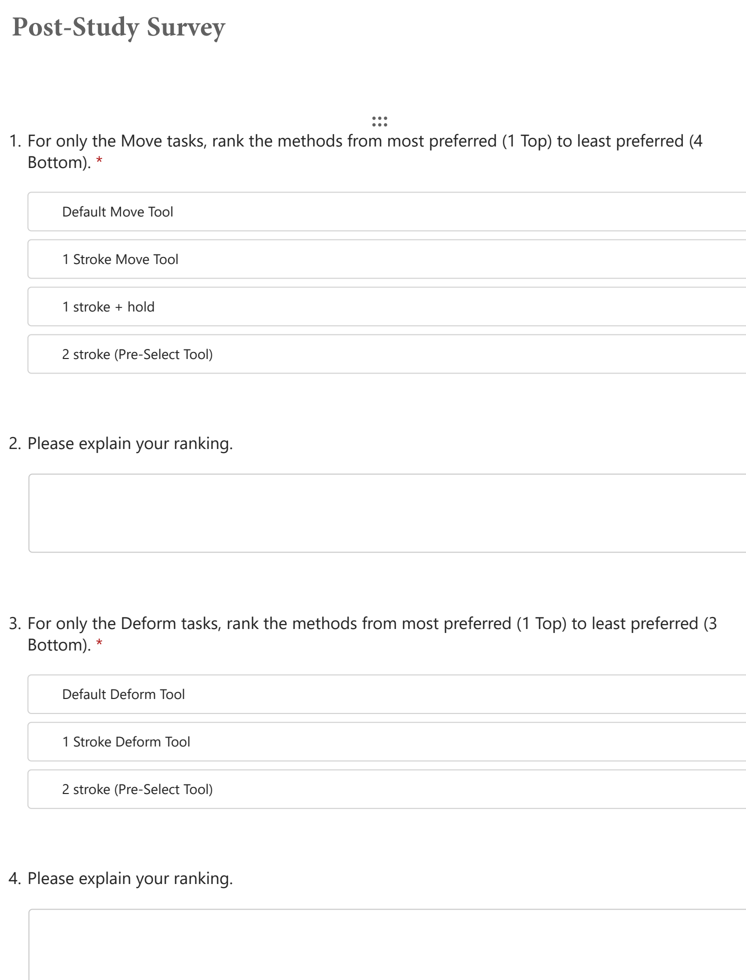
5. Effort: How hard did you have to work to accomplish your level of performance? *

Very Low	Low	Medium	High	Very High
○	○	○	○	○

6. Frustration: How insecure, discouraged, irritated, stressed, and annoyed were you? *

Very Low	Low	Medium	High	Very Hig
○	○	○	○	○

Figure 20: Questionnaire administered after all trials for a `Task` × `Technique` were completed.



Post-Study Survey

1. For only the Move tasks, rank the methods from most preferred (1 Top) to least preferred (4 Bottom). *
   - Default Move Tool
   - 1 Stroke Move Tool
   - 1 stroke + hold
   - 2 stroke (Pre-Select Tool)

2. Please explain your ranking.

3. For only the Deform tasks, rank the methods from most preferred (1 Top) to least preferred (3 Bottom). *
   - Default Deform Tool
   - 1 Stroke Deform Tool
   - 2 stroke (Pre-Select Tool)

4. Please explain your ranking.

Figure 21: Questionnaire administered after all trials for a `Task` × `Technique` were completed.