Quantifying Object- and Command-oriented Interaction

Alix Goguey, Julie Wagner, Géry Casiez

Quantifying Object- and Command-oriented Interaction

Alix Goguey[1], Julie Wagner[2], Géry Casiez[3]

[1]Inria, France, [2]Human-Computer Interaction Group, University of Munich (LMU), Germany,
[3]University of Lille, France
`alix.goguey@inria.fr, julie.wagner@ifi.lmu.de, gery.casiez@univ-lille1.fr`

Abstract. In spite of previous work showing the importance of understanding users' strategies when performing tasks, i.e. the order in which users perform actions on objects using commands, HCI researchers evaluating and comparing interaction techniques remain mainly focused on performance (e.g. time, error rate). This can be explained to some extent by the difficulty to characterize such strategies. We propose metrics to quantify if an interaction technique introduces a rather object- or command-oriented task strategy, depending if users favor completing the actions on an object before moving to the next one or in contrast if they are reluctant to switch between commands. On an interactive surface, we compared Fixed Palette and Toolglass with two novel techniques that take advantage of finger identification technology, Fixed Palette using Finger Identification and Finger Palette. We evaluated our metrics with previous results on both existing techniques. With the novel techniques we found that (1) minimizing the required physical movement to switch tools does not necessarily lead to more object-oriented strategies and (2) increased cognitive load to access commands can lead to command-oriented strategies.

Keywords: interaction sequence; task strategy; metric; theory; finger identification; finger specific

1 Introduction

In HCI research, we sometimes face the problem that two designed interaction techniques might differ in various factors that we cannot control in experiments: individual techniques might require different implementations (*vision-based* hand- vs. *capacitive* touch tracking), different body parts for interaction (e.g. *uni-* vs. *bimanual*) or different modalities (*touch* vs. *mid-air* gestures). For such techniques, comparing performance time becomes either meaningless or does not reveal the exact reasons for the time benefit.

Playing around with the context, Mackay [7] compared floating palette, marking menu and Toolglass [5] when performing two tasks requiring participants to copy or modify Petri-nets. She concluded that the optimal interaction technique in terms of performance varied depending on the task, the user's cognitive context and individual preferences. She further observed that floating palette and marking menu favor *tool-by-tool* actions (*e.g.* first creating all *triangles*, then all *circles*) while Toolglass favors frequent switch between tools.

We believe that interaction techniques – the integration of *physical* and *logical* device design [2] – affects how people solve a task; and that exclusive time metrics do not help researchers in understanding *why* one technique performs faster than another.

We propose additional metrics to help categorizing interaction techniques by automatically and objectively labeling strategies. We believe that used in an iterative development, they would give insight to designers on whether or not their system leads users to adopt an effective strategy for a given task and helps them choosing one interaction technique or another. We compared two techniques from the literature, Fixed Palette [1] and Toolglass [5], with two novel techniques using finger identification on interactive surfaces, *Fixed Palette with Finger Identification* and *Finger Palette* in a vector drawing task. We found that we can correctly conclude the previously identified results [1,7] from our metrics: *Fixed Palette* is a highly command-oriented and *Toolglass* a highly object-oriented technique. We found that *Fixed Palette using Finger Identification* is significantly more object-oriented than *Fixed Palette*. *Finger Palette* and *Fixed Palette* are equally tool-oriented. We discuss cognitive reasons for these differences in strategies. specific

2 Related Work

Appert *et al.* [1] and Mackay [7] define a strategy as the order of elementary actions on objects to solve a task. Both works studied performance of interaction techniques in different contexts and identify which kind of strategy is best suited for each. With the *Complexity of Interaction Sequences* model (CIS), Appert *et al.* take the analyzed structure of an interaction technique and predict its performance time for a given strategy. The strategy should therefore be determined in advance. Mackay did not impose a strategy. Instead she observed interaction sequences and labeled them. Labeling is a tedious task, subjective and error prone considering sequences of actions scarcely belong to one category or the other. Appert *et al.* and Mackay's results concurred: fixed palettes are command-oriented, meaning that users repeatedly re-issue the same command to perform the task while marking menus and toolglasses are object-oriented, meaning that users issue multiple commands with respect to a single graphical object on screen.

Bhavnani and John [3,4] studied higher level strategies (*i.e.* strategies that differentiate novices from expert users) and how users gain expert knowledge. They argue that users need to learn strategies: knowledge of a task and knowledge of tools are not sufficient to make users more efficient with a complex computer application. Cockburn et al. [6] discuss in their review paper various systems that help users to learn better strategies. *Skillometer* [8] is one of these systems helping users to use keyboard shortcuts instead of time-consuming menu navigation. Our metrics are intended to measure lower level strategies as Appert, Beaudouin-Lafon and Mackay studied.

Mackay [7] also measured the average number of identical actions performed before switching to another command: a high score indicates a command-oriented pattern while a low score suggests an object-oriented pattern. Besides being a subjective choice, switching a lot between commands does not necessarily involve being object-oriented (*e.g.* drawing a circle, then a triangle, then filling the circle in blue and finally filling the triangle in red is neither object- nor command-oriented). *Object-oriented* and *command-oriented* strategies are orthogonal to each other. The metrics we introduce are intended to measure automatically the degree of which an interaction sequence is object-oriented and command-oriented. Furthermore, our metrics also allow us to be more ecological since we do not impose users to follow any strategy.

3 Metrics

With a given interaction technique, users might optimize efficiency and perform a compound task using strategies varying between strictly *command-oriented* or *object-oriented*. A strategy (S) can be decomposed in *n* elementary actions (a_i) performed on interactive objects *Obj(a_i)* (the object modified during action a_i). For example, drawing two blue rectangles can be decomposed in the actions of *creating rectangle* (c_{rect}), and *blue-filling* (f_{blue}) performed on two rectangle objects R_1 and R_2: with a *command-oriented* strategy, users are reluctant to switch commands which would result in, *e.g.*, the following sequence: $(c_{rect})_{R1} (c_{rect})_{R2} (f_{blue})_{R1} (f_{blue})_{R2}$; with an *object-oriented* strategy, users favor completing an object before continuing with the next one, which would result in, *e.g.*, $(c_{rect})_{R1} (f_{blue})_{R1} (c_{rect})_{R2} (f_{blue})_{R2}$.

3.1 Quantifying Object-oriented Strategy

With an object-oriented strategy, users finish all their actions on an object before moving to the next one. Therefore we penalize any action occurring on objects previously edited or created. For a Strategy $S = \langle (a_1)_{Obj(a1)}, \dots (a_n)_{Obj(an)} \rangle$ of *n* actions, we measure the *ObjectOriented(S)* ratio as follows:

$$P(S) = \sum_{i=3}^{n} \begin{cases} 1 \text{ if } Obj(a_i) \neq Obj(a_{i-1}) \\ \quad \text{and } \exists j \in [\![1; i-2]\!] \text{ such as } Obj(a_i) \neq Obj(a_j) \\ 0 \text{ otherwise} \end{cases} \quad (1)$$

$$ObjectOriented(S) = ObjOri(S) = 1 - \frac{P(S)}{n - m} \quad (2)$$

If users complete their actions on an object before moving to the next one, *P(S)=0* and *ObjOri(S)=1*. At the opposite if they switch to a different object for each of their action *P(S)=n-m* (with *m* the number of objects on the canvas) and *ObjOri(S)=0*.

3.2 Quantifying Command-oriented Strategy

With a command-oriented strategy, users keep using the same command as long as they can before switching to another one. As a result we penalize any switch to a command previously used. For a strategy $S = \langle (a_1)_{Obj(a1)}, \dots (a_n)_{Obj(an)} \rangle$ of *n* actions, we measure the *CommandOriented(S)* ratio as follows:

$$P(S) = \sum_{i=3}^{n} \begin{cases} 1 \text{ if } a_i \neq a_{i-1} \\ \quad \text{and } \exists j \in [\![1; i-2]\!] \text{ such as } a_i = a_j \\ 0 \text{ otherwise} \end{cases} \quad (3)$$

$$CommandOriented(S) = CmdOri(S) = 1 - \frac{P(S)}{n - c} \quad (4)$$

If users keep using the same command before switching to the next one, *P(S)=0* and *CmdOri(S)=1*. At the opposite, if they keep switching from a command to anoth-

er at each action, *P(S)=n-c* (with *c* the total number of commands used on objects) and *CmdOri(S)=0*.

4 Experiment

To evaluate our metrics, we compared two novel interaction techniques, *Fixed Palette using Finger Identification* and *Finger Palette*, with *Fixed Palette* and *Toolglass*.

4.1 Participants

12 volunteers (3 female, mean age 26) participated in our study. Four reported their hand dexterity as 'good' and height as 'normal'. All were familiar with touch-screen technology and drawing applications.

4.2 Procedure and Tasks

We ran a 4 TECHNIQUE x 3 TASK within-subject design counter-balanced by TECHNIQUE. Unique conditions were repeated 7 times (4 x 3 x 7 = 84 data points per participant) and the order of TASK x REPETITION was pseudo-random. Participants were instructed to optimize time and TASK was to match the position, shape and color of several shapes displayed full-sized with light transparency on the canvas. Figure 1 illustrates the 3 TASKS: each contained 6 objects arranged in a two rows and three columns grid. TASKS contained either objects of same shape and fill color (T1), three shapes and colors spatially grouped (T2) and ungrouped (T3). All TASKS required the same number of actions in order to complete. All techniques provided access to *square*, *circle*, and *triangle* tools and *red*, *green*, *blue* coloring tools. We added an 'erase' tool to correct errors. We intentionally left out logical tools such as 'copy' + 'paste' or 'select group' to avoid noisy data.

We displayed visual cues in the background image that enabled participants to draw all objects without the need for positioning them: the shapes were created by dragging a bounding box; a 15 mm (approximately the width of a finger) tolerance area at each corner of a shape indicated where each drag should start and end; the shape's stroke color turned red when it overshot the tolerated area. Newly created shapes did not have a fill color. When the right color was applied, the shape's stroke color turned green indicating successful completion of the object.
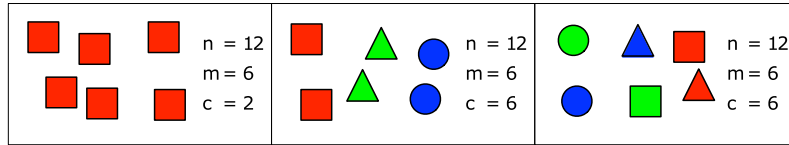

Fig. 1: Examples of instances for TASKS T1, T2 and T3.

We implemented two techniques from the literature: *Fixed Palette*— expected to favor command-oriented strategies, and *Toolglass*— expected to favor object-oriented strategies [7,1]. In addition, we implemented two novel techniques (*Fixed Palette using Finger Identification* and the *Finger Palette*) that we expected would favor object-oriented strategies.

TECHNIQUE 1: **Fixed Palette** *Fixed Palette, a.k.a.* tool palette, is a single-pointer widespread technique (Figure 2a) [1]. It contains a set of commands that users select by pressing the appropriate button. Users conceptually hold the selected tool until they select another one. Since tool-switching requires large movements between canvas and palette, we expect users to follow a command-oriented strategy. We implemented the *Fixed Palette* to remain fixed at the right side of the display.

TECHNIQUE 2: **Fixed Palette using Finger Identification** We extended *Fixed Palette* to a single-handed multi-pointer technique. The onscreen representation remains the same. Users can temporarily assign tools to each finger of their dominant hand: by touching *e.g.*'rectangle' with the index and 'circle' with the middle finger, both tools can be instantly operated using the corresponding finger. Since switching between a limited number of tools (5 fingers max) is quicker than for *Fixed Palette*, we expect to find object-oriented strategies.

TECHNIQUE 3: **Toolglass** The *Toolglass* is a bimanual dual-pointer technique: a widget containing a set of *semi-transparent* buttons [5] is positioned onscreen using the non-dominant hand. Command selection is performed using the dominant hand (Figure 2b). The non-dominant hand's index finger positions the main *Toolglass* containing the six tools, the middle finger positions a second *Toolglass* containing the eraser. Since applying the same tool twice or switching tools requires equal 'effort', we expect to find object-oriented user strategies.

TECHNIQUE 4: **Finger Palette** The *Finger Palette* is a bimanual multi-pointer technique. The non-dominant hand controls the temporal but fixed assignment of tools to fingers of the dominant hand: for a right-handed person, *e.g.*, holding the left index finger down assigns rectangle, triangle and ellipse to the right index, middle and ring fingers (Figure 2c). Tools are applied by the right hand's fingers independent of the left-hand's position. To reveal finger-command mappings, we display a cheat sheet next to the left index finger. We organized color and drawing tools into thumb and index finger palette; we placed the eraser into the middle finger palette. Again, we expect this technique to favor object-oriented strategies, since all commands are directly available from anywhere on the canvas.

4.3 Apparatus

We used an horizontal 32'' 3M touchscreen[1] (Figure 3 left). We merged fingers' on-screen touch position with the 3D position reported by 5 GameTrak[2] devices (Figure 3 right) attached to each fingertip via cords. We wrote a C++ software using the *libgametrak*[3] library, that establishes a correspondence between the tracked finger positions and multiple touch points registered on the multi-touch surface. It uses a homography for each finger. The homographies are determined by a calibration procedure in which 3D points are sampled at known positions in the display reference frame. Once the system is calibrated, the software associates to an onscreen touch the identification of the closest finger.

---

[1] http://www.3m.com
[2] http://en.wikipedia.org/wiki/Gametrak
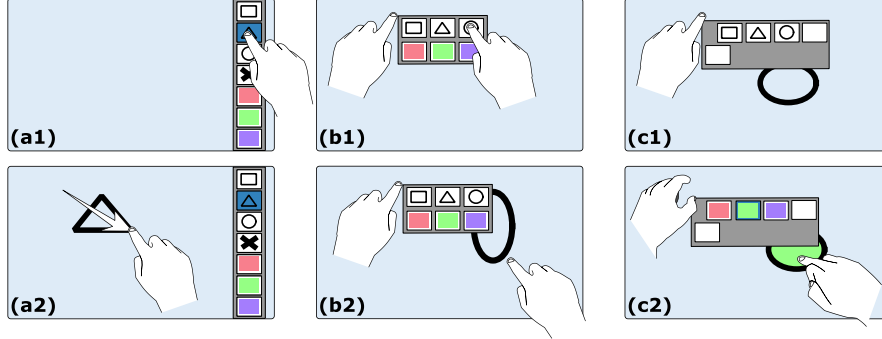[3] https://github.com/casiez/libgametrak/

Fig. 2: Illustrating: *Fixed Palette*, the user selects the triangle tool (a1) and creates a triangle by dragging (a2); *Toolglass*, user positions semi-transparent widget using the non-dominant hand (b1) and starts drawing by press-and-drag through the ellipse button the dominant index (b2); *Finger Palette*, the left hand controls the assignment of tools to the right fingers (c1), user invokes color tools using the left thumb and colors an ellipse green using the middle finger (c2).

Fig. 3: Experimental setup: (left) participant completing TASK T1 using the *Finger Palette*; **(right)** the 5 GameTrak devices located above the 32'' 3M touchscreen.

4.4 Results and Discussion

The dependent variables were the *CmdOri* and *ObjOri* ratios. A one-way ANOVA showed no effect of REPETITION on *CmdOri* and *ObjOri* ratios suggesting there was no learning effect. A repeated-measures MANOVA showed a significant main effect of TECHNIQUE ($F_{6,66} = 10.561, p < 0.0001$) and a significant TECHNIQUE x TASK interaction ($F_{12,132} = 5.201, p < 0.0001$) on *CmdOri* and *ObjOri* ratios (Figure 4).

Metric Evaluation Post-hoc analysis showed significant differences ($p < 0.03$) between all techniques except *Fixed Palette* and *Toolglass*. Figure 4 shows the distribution of both ratios per TECHNIQUE. Analog to previous findings [1,7], participants performed identical tasks either command-oriented when using *Fixed Palette* (*CmdOri* ratio: $\overline{m} = 0.99, CI[0.99,1.00]$ and *ObjOri* ratio: $\overline{m} = 0.05, CI[0.04,0.07]$, $\overline{m}$ is the mean) or object-oriented using *Toolglass* (*CmdOri* ratio: $\overline{m} = 0.51, CI[0.46,0.55]$ and *ObjOri* ratio: $\overline{m} = 0.68, CI[0.63,0.74]$). This result provides a first validation of our metric.

Command-oriented strategies with *Finger Palette* *CmdOri* and *ObjOri* ratios are not significantly different for *Finger Palette* (*CmdOri* ratio: $\overline{m} = 0.93, CI[0.91,0.96]$

and *ObjOri* ratio: $\overline{m}$ = *0.06, CI[0.04,0.08]*) and *Fixed Palette*: the smaller physical movement required to switch tools using *Finger Palette* did not affect users' choice to adopt an object-centered strategy. We hypothesize that this is due to the tool grouping of this technique that might encourage a command-centered strategy.

Task-dependent strategy with *Fixed Palette using Finger ID* For *Fixed Palette using Finger Identification*, we found significant differences ($p < 0.05$) between TASK: users adopted a significantly more object-oriented strategy with T1 (*CmdOri* ratio: $\overline{m}$ = *0.61, CI[0.51,0.71]* and *ObjOri* ratio: $\overline{m}$ = *0.44, CI[0.35,0.53]*, see yellow dot in Figure 4) than in both other tasks: T2 and T3 (*CmdOri* ratio: $\overline{m}$ = *0.92, CI[0.90,0.95]* and *ObjOri* ratio: $\overline{m}$ = *0.14, CI[0.10,0.18]*, little yellow circle in Figure 4). In T1, that consisted in drawing only red rectangles, participants reported that the personalization of command-finger mappings facilitated memorization. With increasing diversity of shapes and colors, memorizing the mappings became more difficult leading to a command-oriented strategy.
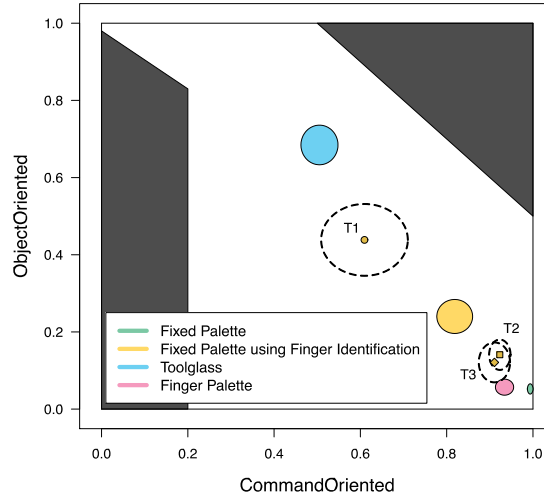
Fig. 4: Mean *ObjOri* and *CmdOri* ratios for each TECHNIQUE. Ellipses represent the 95% confidence interval for the means. The gray areas represent the unreachable areas for the tasks we considered. The yellow dot corresponds to the mean strategy used in T1 using *Fixed Palette using Finger Identification* and the yellow square and diamond correspond to the mean values for T2 and T3, illustrating the interaction effect.

5 Conclusion and Future Work

We introduced two novel measurements that, combined together, can help researchers in quantifying the effects of interaction techniques on interaction sequences (users' strategy) when solving a task. Our metrics together penalize both the number of tool switches and switching the focus between on-screen objects. We compared four techniques to measure users' strategy on three types of drawing tasks.We empirically replicated previous results found regarding *Fixed Palette* and *Toolglass*, validating

our metric [1,7]: users follow a command-centered strategy using *Fixed Palette* and an objectcentered strategy using *Toolglass*.

For *Finger Palette*, we found that users follow a command-oriented strategy. We conclude that techniques minimizing required physical movements to switch tools do not necessarily lead to more object-oriented strategies. We hypothesize for future research that the organization and grouping of commands in the interface has an effect on the choice of strategy. We found that, for our task, people significantly favor object-oriented strategies when using *Fixed Palette using Finger Identification* compared to *Fixed Palette* in tasks with low tool diversity. High tool diversity leads to reported cognitive load of remembering command-finger mappings. This finding suggests that promoting object-oriented interaction tools should not only minimize physical movements, but cognitive aspects as well.

As future work, we plan to adapt our metric to support the two limitations at present: (1) We seek to investigate our metrics with real-world tasks, where users do not necessarily know the final outcome of a task in advance. (2) We seek to adapt our metrics to higher-level tool concepts. We applied our metric to investigate the effect of interaction techniques on interaction sequences. Tasks could also be solved using higher-level logical tool concepts, *e.g.* `copy-and-paste`, as investigated by Bhavnani *et al.* [4].

References

1. Appert, C., Beaudouin-Lafon, M., Mackay, W.: Context matters: Evaluating interaction techniques with the cis model. In: Fincher, S., Markopoulos, P., Moore, D., Ruddle, R. (eds.) People and Computers XVIII — Design for Life, pp. 279–295. Springer London (2005), `http://dx.doi.org/10.1007/1-84628-062-1_18`
2. Beaudouin-Lafon, M.: Instrumental interaction: An interaction model for designing post-wimp user interfaces. In: Proc. of CHI'00. pp. 446–453. ACM, `http://doi.acm.org/10.1145/332040.332473`
3. Bhavnani, S.K., John, B.E.: Delegation and circumvention: Two faces of efficiency. In: Proc. of CHI'98. pp. 273–280. CHI '98, `http://dx.doi.org/10.1145/274644.274683`
4. Bhavnani, S.K., John, B.E.: The strategic use of complex computer systems. Hum.-Comput. Interact. 15(2), 107–137 (2000), `http://dx.doi.org/10.1207/S15327051HCI1523_3`
5. Bier, E.A., Stone, M.C., Pier, K., Buxton, W., DeRose, T.D.: Toolglass and magic lenses: the see-through interface. In: Proc. SIGGRAPH'93. pp. 73–80. ACM, `http://doi.acm.org/10.1145/166117.166126`
6. Cockburn, A., Gutwin, C., Scarr, J., Malacria, S.: Supporting Novice to Expert Transitions in User Interfaces. ACM Computing Surveys (CSUR) 2014, 47(2), 31. `http://doi.acm. org/10.1145/2658850.2659796`
7. Mackay,W.: Which interaction technique works when? Floating palettes, marking menus and toolglasses support different task strategies. In: Proc. AVI'02. pp. 203–208. ACM, `http://doi.acm.org/10.1145/1556262.1556294`
8. Malacria, S., Scarr, J., Cockburn, A., Gutwin, C., Grossman, T.: Skillometers: Reflective widgets that motivate and help users to improve performance. In: Proc. UIST'13. pp. 321–330. ACM, `http://doi.acm.org/10.1145/2501988.2501996`