



HAL
open science

Interaction Illustration Taxonomy: Classification of Styles and Techniques for Visually Representing Interaction Scenarios

Axel Antoine, Sylvain Malacria, Nicolai Marquardt, Géry Casiez

► To cite this version:

Axel Antoine, Sylvain Malacria, Nicolai Marquardt, Géry Casiez. Interaction Illustration Taxonomy: Classification of Styles and Techniques for Visually Representing Interaction Scenarios. CHI 2021 - ACM Conference on Human Factors in Computing Systems, May 2021, Yokohama, Japan. 10.1145/3411764.3445586 . hal-03137867

HAL Id: hal-03137867

<https://hal.science/hal-03137867>

Submitted on 10 Feb 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Interaction Illustration Taxonomy: Classification of Styles and Techniques for Visually Representing Interaction Scenarios

Axel Antoine

Univ. Lille, CNRS, Inria, Centrale Lille, UMR 9189 CRISTAL
Lille, France
axel.axantoine@univ-lille.fr

Nicolai Marquardt

University College London
London, UK
n.marquardt@ucl.ac.uk

Sylvain Malacria

Univ. Lille, Inria, CNRS, Centrale Lille, UMR 9189 CRISTAL
Lille, France
sylvain.malacria@inria.fr

Géry Casiez*

Univ. Lille, CNRS, Inria, Centrale Lille, UMR 9189 CRISTAL
Lille, France
ger.y.casiez@univ-lille.fr

ABSTRACT

Static illustrations are ubiquitous means to represent interaction scenarios. Across papers and reports, these visuals demonstrate people’s use of devices, explain systems, or show design spaces. Creating such figures is challenging, and very little is known about the overarching strategies for visually representing interaction scenarios. To mitigate this task, we contribute a unified taxonomy of design elements that compose such figures. In particular, we provide a detailed classification of *Structural* and *Interaction* strategies, such as composition, visual techniques, dynamics, representation of users, and many others – all in context of the type of scenarios. This taxonomy can inform researchers’ choices when creating new figures, by providing a concise synthesis of visual strategies, and revealing approaches they were not aware of before. Furthermore, to support the community for creating further taxonomies, we also provide three open-source software facilitating the coding process and visual exploration of the coding scheme.

CCS CONCEPTS

• **Human-centered computing** → **Human computer interaction (HCI); Visualization theory, concepts and paradigms.**

KEYWORDS

interaction illustrations; taxonomy; figures; interactive scenarios

ACM Reference Format:

Axel Antoine, Sylvain Malacria, Nicolai Marquardt, and Géry Casiez. 2021. Interaction Illustration Taxonomy: Classification of Styles and Techniques for Visually Representing Interaction Scenarios. In *CHI Conference on Human Factors in Computing Systems (CHI '21)*, May 8–13, 2021, Yokohama, Japan. ACM, New York, NY, USA, 22 pages. <https://doi.org/10.1145/3411764.3445586>

*Also with Institut Universitaire de France.

1 INTRODUCTION

Interaction scenarios describe situations when one or multiple users interact with one or more interfaces of digital computing systems or devices. When interested in one of these scenarios, the ideal way to understand the interactivity is to experience it, or at least to be able to see it in action. Unfortunately, interacting with the system is not always possible. Indeed, the main medium to communicate about research in Human-Computer Interaction remains research papers, that use non-dynamic formats (in particular PDF or static HTML web pages). As “*a picture is worth a thousand words*”, authors often design and represent static graphical illustrations to explain these interactive scenarios, most of the time associated with captions or legends to form a complete figure [27]. Furthermore, an increasing number of journals recommend adding graphical abstracts [38] summarising the submissions (for example, Elsevier journals suggest including visual abstracts [25]). The figure should give a *clear representation of the work described in the paper, summarise the content* to help readers quickly gain and understand the main take-home message of the paper, encourage browsing, promote and identify research papers.

Illustrations are not only a useful for describing interactive scenarios in research papers, but they are also widely used as effective visual means in presentations or for communicating ideas during meetings. They can take many forms and represent various information that can be photographs, drawings, diagrams depicting concepts, and ideas or charts helping to visualise data. The structure of the illustrations might also vary, using one or multiple frames, being augmented with titles, labels, and even the nature of the captions can be considered as part of the illustration [27].

While ubiquitous and widely used, little is known regarding the different approaches to create such static illustrations and existing works on this topic are limited in terms of domain focus (*e.g.* design, education) or interaction context (*e.g.* gestures). In this paper, we unpack a rich taxonomy of styles and techniques that unifies works from the literature, and investigates how the HCI community uses static figures as a way to depict interactive scenarios. We call *static figures* any visual representations that are used on static forms of media (such as PDF papers) containing an illustration (*e.g.* drawing, photograph) with additional information (*e.g.* caption, legend, title). We consider in this work only figures representing interactive contexts, where the interaction can be seen as *Tool Use* [36], *i.e.* figures “*defining how the user acts with a system and how the system acts with*

a user”, as they can be particularly complex to design and make use of elaborate illustration techniques to represent specific aspects of the interaction (3D, perspective, dynamic gestures, timing, user’s body, etc.).

First, we propose a taxonomy of static figures that illustrate tool use, where we categorise figures according to different design elements (conceptual or visual attributes of an illustration used to encode the interaction [64]) they are composed of. We classify the design elements under two main categories, the *what* design elements (that characterise the concepts represented in the figure) and the *how* design elements (that characterise the visual properties). Then, we coded design elements for each image of a dataset extracted from the proceedings of four major ACM 2018 conferences in HCI (CHI, UIST, CSCW and Ubicomp) and from a set of patents about interaction designs. Finally, we analyse the coded design elements and corresponding figures to reveal different strategies used to represent interaction.

An increasing number of papers in HCI are published every year, resulting in an increasing number of graphical illustrations being produced. Yet, no guidelines or recommendations exist to help researchers create these kind of illustrations. The strategies identified in this paper provide an overview of the figures produced by HCI researchers. It can help early-career researchers working on visual representations of their work, and foster reflection of senior researchers on their own (previously) created figures.

This paper makes the following main contributions:

- (1) A unified taxonomy of design elements that compose figures representing interactive scenarios. Our taxonomy integrates existing taxonomies and synthesises a broad spectrum of techniques and approaches across the many different interactive scenarios illustrated in the HCI community.
- (2) A set of *Structural* and *Interaction* strategies in existing figures used to represent specific aspects of the interaction.
- (3) Three open-source software tools: one application to facilitate the time-consuming and tedious coding process, and two on-line tools to explore the taxonomy we created and identify strategies. These tools can be used to repeat our methodology either with a different dataset or driven by different goals in order to extract novel strategies.

2 RELATED WORK

2.1 Existing taxonomies

Our work builds upon and extends related taxonomies that classify visual design elements, all with different and specific motivations, exploring either the impact of illustrations on learners’ behaviour [27], new product development [68], gesture representations [64] or trace figures [4]. To the best of our knowledge, our work is the first taxonomy focused on the classification and characterisation of illustrations that depict interactive scenarios. In this section we summarise the most relevant visual taxonomies, and their relation to the work presented in this paper.

Pei *et al.* proposed a taxonomy of visual design representations in the context of product development [68]. In particular, they discuss that in the field of product design, the drawing style might be different depending on the life cycle of an illustrated product or the purpose conveyed through the illustration. Using a dataset

built by designers and engineers, their taxonomy divides the visual design representations in four main categories: *sketches*, *drawings*, *models* and *prototypes*. While this taxonomy is focused on specific design representations, we believe their work is of interest to our larger exploration of interactive scenario illustrations, as interactive systems might require different types of representation as well. More specifically, Pei *et al.* propose a specific sub-category: 2D Visual Design Representations > Drawings > Industrial Design Drawings > Scenario and Storyboard, where the purpose of such illustrations is to *suggest user and product interaction and portray its use in the context of artefacts, people and relationships*.

Another taxonomy can be found in McAweeney *et al.*’s work [64]. They first conducted an elicitation study with designers and researchers to understand the processes and tools used to create gesture representations. The elicitation study pointed out that no guidelines existed to assist researchers in designing gesture representations. Then, the authors constituted a dataset from 30 papers published in ACM conferences (CHI, ISS and Ubicomp), including trace figures, photographs, computer graphics, abstract lines, dots and texts. Using information from the elicitation study, they coded the dataset and classified the identified design elements in six dimensions grouped in two main categories: *structural* and *details*. *Structural dimensions* (*Perspective*, *Frame* and *Colour*) are described as necessary dimensions to design any representation while *Details dimensions* (*Body Context*, *Environmental Context* and *Gesture Elements*) are described as optional dimensions commonly used to extend/enrich the structural representation of the gesture.

More recently, Antoine *et al.* proposed a taxonomy of *trace figures*, a specific type of illustrations they define as “*graphical representations of the most essential features of a scene by using contours/outlines of objects, people and the environment*” [4]. They extracted 124 trace figures from the 222 papers published in the 2015-2017 ACM UIST conference proceedings and used two coders to code the dataset. They identified five categories of trace figures (*demonstration of gestures*, *overview of system setup or assembly*, *interaction sequences*, *design space illustrations* and *others*) and they extracted eight characteristics depicted (*person’s body*, *hands or fingers*, *devices and objects*, *screen user interfaces*, *environment*, *use of colour*, *annotations*, *static vs dynamic* and *use of perspective*). While this taxonomy covers a number of different interactive scenarios used in the HCI community, it only contains a brief description of identified categories and characteristics. Moreover, the work is focused on trace figures only and, as such, excludes figures based on photographs that are likely to be used to illustrate other scenarios (typically too complicated to reproduce as trace figures) and using different strategies (that would not be adapted to trace-based drawings).

Finally, while different from figures illustrating interacting scenarios, Fleming’s taxonomy [27] relates to our visual taxonomy. He analysed 787 illustrations extracted from 40 textbooks from four subject areas: English, History, Mathematics and Science [27]. The purpose of the study was to observe the impact of illustrations on learners’ behaviour. To do so, he established a taxonomy of instructional illustrations by tagging each illustration with attributes grouped into 11 scales for a total of 107 categories. The scales were as follows: *Area*, *Framing*, *Shape*, *Position*, *Elements*, *Chroma*, *Achroma*, *Encoding Style*, *Encoding Medium*, *Information Level* and

Unification. The scales of this work, and the general discussion of visual representations by Fleming, informed our classification scheme discussed shortly.

2.2 Bridging existing taxonomies with theories on illustration

As explained above, very few taxonomies specifically targeted at illustrations exist, and they remain limited either in terms of context of interaction, type of figures or domains of interest. Nevertheless, these taxonomies contain information that can be merged together towards a larger and more holistic taxonomy of illustration of interactive scenarios. In the following sections, we link the taxonomies presented in the previous subsection with other works from the literature, typically design books or articles, to analyse how design elements present in figures can support the representation of interactive scenarios.

2.2.1 Framing interactive scenario illustrations.

A first characteristic of illustrations lies in how they are structured, for instance how many images are used, how they are framed and the layout they are composed with.

Notably, Buxton argues that “*sketches for experience and interaction design [...] differ from conventional sketching since they have to deal with time, phrasing and feel*” [14]. As such, the representation of an interaction scenario should respect different attributes, for instance *time* (since interaction might be ongoing), *clear vocabulary* and *distinct gesture* to only represent the essential elements that compose the interaction, *minimal detail* and *appropriate degree of refinement* to minimise the reader disturbance. These attributes relate to similar concepts mentioned in McAweeney *et al.*'s taxonomy of gesture representations [64] where they mentioned five important aspects: *Time, Position, Posture, Motion and Touch*.

Similarly, while the *Details* dimensions by McAweeney *et al.* [64] describe the addition of precise details to understand different aspects of the represented gesture, the *Structural* dimensions are necessary to design any kind of interaction illustration. Specifically, the *Frame* dimension (which is close to the *Framing* scale presented in Fleming's taxonomy [27]) mentions that illustrations can have one or multiple frames to structure the representation [64].

A notion similar to framing is discussed in *Understanding Comics*, Scott McCloud's book exploring formal aspects of comics and their fundamental vocabulary [65]. More precisely, McCloud discusses the notion of *Panels* that can be used to “*fracture both time and space, offering a jagged rhythm of unconnected moments*” (p67), and use *closure* (i.e. space between panels) as ways to “*connect these moments*”. Such transitions between the different frames of a comics storyboard can have different forms (p.70) [65], where few of them might be relevant in an interactive scenario illustration. Notably, McCloud introduces *moment-to-moment* as a frame transition between two very close moments in time (that in the context of HCI could be relevant to illustrate transitions between two UI captures of a system), *action-to-action* as a frame transition between two close actions in time of a same subject (e.g. a user pointing at a specific target), *subject-to-subject* as frame transition between different subjects (e.g. showing a data transfer between two collaborating users) and *aspect-to-aspect* as a frame transition between different

aspects of the represented context (e.g. giving different points of view of a prototype system device).

Greenberg *et al.* also identified the importance of frames and their transitions when describing an interactive context, as they are “*what triggers a change in state*” and can be “*triggered by one or more user actions*” [31]. A good example is UI transitions, which are often represented in HCI papers to illustrate the step-by-step interaction with an interface.

Finally, an interesting structural dimension is *perspective*, a dimension for the point of view of the representation. McAweeney *et al.* listed five different points of view: *1st person, 3rd person, mirror, bird's eye* and *side angle* [64]. Note that frame and perspective are dependent as perspective encompasses the point of view used in a frame.

2.2.2 Illustrate dynamic interactions through the medium of static images.

In illustrations representing dynamic, motion can be explicit (an object moves within one frame) or implicit (an object moves between two different frames) [65], especially when representing step-by-step operations that are generally illustrated as storyboards or interaction sequences [4]. Designing a storyboard consists in structuring a figure with multiple frames, dependent from each other, and indicating a continuous flow of events in time and/or space to represent the expected behaviour. Greenberg *et al.* [31] said that “*storyboard is a collection of sketches, where each sketch is a key frame or a state that portrays an important moment in the interface sequence*”. Multi-step operations can also be structured as a single illustration, using key structure elements to indicate in which order to read the elements of a figure (typically numbers). In the context of instruction manuals, step-by-step instructions have been shown to be preferred by users over a single complex figure (such as an exploded view), to understand assembly instructions [1], thus confirming the usefulness of storyboards for presenting sequential information.

Alternative techniques to storyboards have also been identified by McAweeney *et al.* [64] and Antoine *et al.* [4]. For instance, arrows can be used to represent trajectories or motion paths [4, 64] using a plain line-style on a surface or dotted line in mid-air [64]. Touchpoints [64] (or contact points [4]) can be used to illustrate physical contacts between fingers and a surface. Finally, both mentioned the possibility to use a “ghost” effect [64] (or stroboscopic effect [4]) to illustrate a motion where “*previous states/positions are drawn in different colours or line styles*” [4]. Interestingly, arrows and stroboscopic effects were already identified as best candidates to illustrate dynamic by Cutting in his book on the representation of motion in static images [22], and software tools assisting the production of figures that might illustrate motion usually implement both methods [4, 19].

Variations of arrows and stroboscopic effects are also often used to illustrate dynamic in comics [65]: wave symbols to describe the sound propagation, motions or action lines for movements, trajectory paths, multiple images (a.k.a stroboscopic effect), a streaking effect (using game of shadow/very simplified version of the model) or a blurring effect. Moreover, some of these effects can be applied to the object or to the background whether the camera is moving with the object or not.

2.2.3 Appearance of an illustration.

Other than the structure of an illustration, its overall appearance might also be impacted by the content displayed and how it is displayed. Typically, the aspect is directly impacted by the source material used: a photograph, a screen capture, or a drawing. As previously mentioned, drawing style might differ in the field of product design depending on the life cycle of the illustrated product or the purpose conveyed through the illustration, and belong to one of four different main categories: *sketches*, *drawings*, *models* or *prototypes* [68]. In the specific context of automotive design sketches, Tovey *et al.* studied the drawing process of postgraduate students and professional designers [85]. Their results depicted different categories of lines to represent an object such as *form lines* (e.g. lines defining the object's contours), *crown lines* (e.g. lines used to add volume to the object) or *area lines* (e.g. lines defining the object's shadows).

A more flexible approach is suggested by Scott McCloud who established that the style, the appearance of comics, could be positioned in a triangle that he calls the *pictorial vocabulary* of drawings [65] (Chapter 2, p51-53), and whose corners are *Reality*, *Language* and *The Picture Plane*, respectively corresponding to *realist*, *iconic* and *abstract* styles. Thus, each drawing can be positioned within this triangle depending on the style the author used.

We observe the same philosophy with Wileman (see figure 8.6 in [34]) describing how to represent an object from an abstract concept to realistic drawing. This notion of style and drawing complexity is also depicted by Buxton [14] through the *minimal detail* key element or by Fleming [27] through the *encoding style* category conveying the hypothesis that when using illustrations to illustrate a discourse other than design, the style might need to be simplistic. This is particularly the case in trace figures, where the drawing complexity is very low as only the outlines of the represented objects are drawn.

In the specific context of illustrating interaction, the appearance of an illustration can differ for specific elements. Typically, Antoine *et al.* and McAweeney *et al.* both identified that line style can be used as a way to express specific motion paths, for instance by using dotted lines for mid-air interaction [4, 64].

Another factor important for the appearance of an illustration is the potential use of colours. For both McAweeney *et al.* [64] and Antoine *et al.* [4], the *Colour* dimension is simply a boolean value representing the presence or not of colour in the illustration. They consider colour as a key element and classify the dimension in the *structure* category. For Fleming, however, the use of colours is described through the *Chroma* and *Achroma* categories [27] which imply more details such as transparency, tint and hue.

Note that compared to conventional drawing or photographs, colours can play a critical role in the context of illustrating interaction. Typically, it can be used as a way to focus on specific aspects of an illustration such as emphasising motion [64] or highlighting touch contact points (with coloured circles) or a specific area [4]. Finally, colour might also be used as a coding scheme where different colours can group (respectively distinguish) elements of a same (respectively different) types [18].

2.2.4 Adding additional information and explanations to illustrations.

Fleming considered each figure to be composed of three distinct types of elements [27]: *Pictorial*, *Verbal* and *Design* elements. *Pictorial* elements are the illustrations, *i.e.*, what is represented such as geometric shapes, schematic drawings or graphs; *Verbal* elements are text elements added to the figure such as labels, legends or numbers; and *Design* elements are geometrical elements added to the figure in overlay such as arrows, lines or coloured areas. Such additional information added to the drawing/photography figure itself are part of the illustration and considered to be useful to help the dissemination of important information.

Hansen explored this graphical language and identified graphic tools to help visualisation [34]. He categorised them in six families: circles or curvoids, square-square corners, triangles, lines, points and fuzz or Fuzzy Ideas. This implies visual codes are used to sort and categorise presented information.

To simplify the visualisation of any representation and help the user to rapidly identify key elements from all the different elements structuring an illustration, some annotation techniques may be used. Chi, in her thesis on the design of tools that rely on physical demonstration to produce illustrations of human movements [18], listed such annotation techniques. *Highlights* are used to address attention via shapes (mark the zone) and colours (use a different colour from the background, flashy colours). *Call-outs* are used to present a detailed view while preserving the context (a.k.a magnifiers or zooms). *Text annotations* are used to provide additional useful information that cannot be drawn. *Icons* can be used to indicate the general situation (e.g. valid or invalid situation). *Numbers* can be used to indicate the numbers of a multi-step operation (which can be an alternative to a stroboscopic effect).

2.2.5 Bridging with patent figures.

Interactive context illustrations can also be found in figures that are used in patents describing novel interactive devices or systems. Illustrations are highly recommended for such patents as they provide visual context to make sure that the idea behind the technical aspects of the patent is clear. The creation of a patent figure follows standards [5, 88], from its organisation to its appearance. One important aspect of a patent figure organisation is the layout/page format: figure margins, type of paper, figure and text size are indicated. Regarding the illustration appearance, it has to be a black-and-white drawing but colours and photographs exceptions might be accepted in specific cases. While the drawing should be as clear as possible, *shading* is recommended to give 3D perception to the drawing when needed. Finally, extra information is highly, notably to connect visual elements in illustrations to corresponding text descriptions. For that end, lines must be used to link the visual feature illustrated and the corresponding text using a number identifier, be as short as possible and not cross. Arrows can also be used as extension of these lines to indicate a global area or, as seen before, the direction of a movement. Interestingly, text is not recommended in the illustration unless highly needed.

3 TAXONOMY OF INTERACTIVE SCENARIOS

3.1 Context and objective

The main objective of this work is to propose a novel comprehensive taxonomy of how interaction can be illustrated, that builds on top of existing taxonomies [4, 27, 64, 68] and that is not limited to specific

Composition	Point of view *	1 st person ●		
		3 rd person ●		
		Overshoulder 3/4		
		Top		
		UI only		
	Layout	Number of frames *	One frame ●	
			Multiple frames ●	
		Sub-framing	Inset (PiP)	
			Magnification lense	
			UI embedded ○	
	UI overlay			
	Juxtaposed			
Visual techniques	Emphasize ○	Region ●	Color area ○	
		Element	Arrow	
			Color highlight ○	
			Enclosing	Exact contour line
			Shape	
	Background	Grayed out/Blurred		
		Removed		
	Anonymization	Blur		
		Filling		
	Grouping and linking	Color grouping		
		Identifiers grouping		
		Grouping arrows		
		Text annotation ●		
	Dynamic ●	Effects	Stroboscopic effect ● ○	
			Onomatopoeias	
Motion blur				
Lines, arrows ● ○		Trajectories ○		
		Direction		
Transition				
Projection				
Contact shapes ● ○				
Structure	Identifiers	Numbers ●		
		Letters		
Title				
Measure	Arrows			
	Text indicator			
Interactive Scenario ○ ○	Purpose *	Design space ●		
		Interaction sequence ●		
		Interactive system ●		
	Time* ○	Moving ●		
		Still ●		
	Activity	Production		
		Entertainment		
		Medical		
		Driving		
		2D/3D creation		
Fabrication				
Communication				
Data manipulation				
Users ○ ○ ○	Number	One user		
		Multiple users		
	Body part ● ●	Specific part ●	Hand ○ ●	
			Fingers ○ ●	
			Head ○	
			Foot	
		Lower body ●		
		Upper body ●		
Full body ● ●				

Visual characteristics	Type *	Drawing ●	Realism	Simplistic
				Realistic
		UI ●	Drawing	
			Rendering ○	
		Photo		
		Data visualization ○		
		Clipart/icon		
		Text ● ○ ○		
		Transparency		
	Color * ● ● ● ● ●	Hue ○ ●	Colors ● ● ● ● ●	
Gray-scale				
Monochrome ○ ●				
Line style	Dashed ● ○			
	Width			

Interactive system	Action	Discrete	Pointing		
			Key press ○		
				Symbolic gesture ● ●	
		Continuous	Translating		
			Rotating ○		
	Scaling				
	Scrolling				
	Deforming				
	Writing/drawing				
	Interaction type *	Touch interaction ○ ○			
		Mid-air interaction ○			
		Distal interaction			
		On body interaction			
		Computer interaction			
		Tangible interaction ○ ○			
		Controllers interaction			
		Pen interaction			
	Gaze interaction				
	Input channel *	Tracking based	Outside in ○		
			Inside out		
		Device based ●	Mobile devices ○ ○		
			Large surfaces ○		
			Desktop devices		
			Laptop devices		
			Controllers		
Tangible objects ○					
AR ○					
VR ○					
Sound based	Speech input				
Output modality *	Visual	Large displays			
		Computer displays			
		Mobile displays			
		Projected displays			
		Head mounted displays ○			
	Lights				
Audible	Sound based				
Speech output					
Situation ● ●	Desktop				
	Public/private transport ○				
	Outdoors				
	Private indoors				
	Public indoors				

□ How codes	□ What codes	★ Key Categories
● Explicit equivalent element	○ Mentioned equivalent element	
● Antoine <i>et al.</i> [4]	● Fleming [27]	
● McAweeney <i>et al.</i> [64]	● Pei <i>et al.</i> [68]	

Table 1: Our hierarchy of design elements composed of six main categories. Key categories used to facilitate the coding process are displayed with a star symbol. Existing design elements introduced with previous taxonomies are displayed with a coloured circle. Plain circles show elements composing existing taxonomies while empty circles show elements mentioned in the corresponding papers. We took into account the equivalent terms such as *Dashed line* in our taxonomy versus *Dotted line* in the taxonomy by McAweeney *et al.* [64].

types of interactions [64]. This new hierarchy of design elements covers two fundamental aspects of figures representing interactive scenarios: *what* is represented in the figure (*what codes*) and *how* it is represented (*how codes*). We built a large dataset using figures extracted from four HCI conferences and applied a grounded theory approach [50, section 11.4.1.1] to determine the design elements composing our taxonomy. Our global assumption regarding this work is that identifying relationships between the different design elements used for a figure would help us understand higher-level strategies employed to represent specific types of interaction.

3.2 Data set collection

We collected the 2018 proceedings from the ACM CHI, UIST, CSCW and UbiComp conferences that were chosen as they were the four top conferences in HCI on Google Scholar when this process was initiated, and cover the full spectrum of interactive contexts that can be represented in HCI research. We automatically extracted 7243 figures and corresponding captions from the proceedings using *pdffigure2*¹. Finally, we manually sorted the extracted figures to only keep the 616 figures illustrating interactive scenarios that can be seen as tool use [36], *i.e.* figures involving one or several users interacting with one or several interactive systems, and discarded figures that did not fit this definition (*e.g.* data charts, tables or UML diagrams).

As presented in the related work, we also decided to consider patent figures representing interactive contexts. We therefore complemented our dataset with figures from patents found on *GooglePatent* in the first 30 pages using the keyword “interaction”. We only kept patents including at least one figure representing an interactive scenario, which gave us a total of 47 patents. We then manually extracted 179 figures from these patents.

In the end, our dataset consists of a total of 179+616=795 figures illustrating interactive scenarios, extracted from both ACM proceedings and patents.

3.3 Coding procedure

3.3.1 Step 1: Grounded theory approach. We first randomly extracted from our figure set a subset of 100 figures. As the proportion of patent figures over the whole set is about 22%, we ensured that the same proportion was kept in the extracted subset. Three co-authors of the present paper acted as *subjective* coders in this step². They were instructed to identify the design elements of each figure by creating codes from scratch, with only one constraint: each code had to answer one of the following questions: *What is represented on the figure? (what codes)* and *How is it represented? (how codes)*. Once the figures were coded, all the codes created were analysed. To do that, each code was printed on paper strip using a colour scheme to differentiate *what codes* (red) from *how codes* (blue). Each paper strip⁰ included the name of the code and the number of times each coder used it, giving the information on how significant a code can be regarding its number of occurrences. Coders then manually and collaboratively arranged together the codes in different categories/hierarchies based on the relationships

¹<https://github.com/allenai/pdffigures2>

²As explained below, this step required spatial co-location thus limiting the number of coders who could participate between the codes, their similarities and their significance. This

manual process helped to quickly create a preliminary version of taxonomy.

3.3.2 Step 2: Iteratively refine the taxonomy. The preliminary version of the taxonomy was then refined by iterating on the established codes, this time with four coders (including the three from step 1). Each iteration was composed of the following three actions. First, coders exchanged about the current version of the taxonomy to establish a list of modifications to apply such as adding, removing, merging or reorganising categories. Second, a new subset of figures was extracted (still keeping 22% of patents) that coders were asked to code using the newly modified codes. Third, an agreement score between coders was computed. As recommended by McDonald *et al.* [66, section 5.3.2], we used inter-rater reliability (and more precisely Fleiss’ Kappa and Krippendorff’s Alpha agreement scores, see section 6 for a discussion of how these methods were adapted to our problem) as a metric to verify agreement between coders.

Note that after the first iteration, some codes in certain categories were almost systematically selected. We therefore considered the corresponding categories as *key categories*, meaning that at least one code inside each category should probably be selected (but was not mandatory) when coding a figure. Key categories were *point of view*, *number of frames*, *type*, *colour*, *purpose*, *time*, *interaction type*, *input channel* and *output modality*.

Coders proceeded to four refinement iterations in total, with respective results of agreement scores for each iteration presented in table 2. The first iteration was conducted on a subset of 40 figures to have a consequent dataset to start with. Subsets of only 20 figures were used for the three other iterations. Coders stopped after the fourth iteration, after which 25% of the entire dataset had been coded and agreement scores confirmed as stable and *substantial* [49] suggesting that coders had agreed on the codes and their definitions. Regardless, modifications proposed by coders were systematically minor such as simply renaming a code, or adding a new item to get exhaustive categories (*e.g.* in *activity* or *situation* categories), which are directly dependent on the figure context without providing useful information. Finally, we also computed the average number of codes selected by figures for each iteration. Interestingly, that number increases with the iterations, possibly resulting from both new codes that were added and coders getting more confident across the iterations. A complete overview of the refined taxonomy can be found Table 1. A full version, with code descriptions and definitions, is provided as adjunct material to this submission.

Iteration	#figures	#codes	Fleiss’ κ	Krippendorff’s α
1	40	24.1	0.61 ($\sigma = 0.08$)	0.61 ($\sigma = 0.08$)
2	20	23.6	0.66 ($\sigma = 0.11$)	0.66 ($\sigma = 0.11$)
3	20	25.4	0.61 ($\sigma = 0.10$)	0.61 ($\sigma = 0.10$)
4	20	27.2	0.63 ($\sigma = 0.10$)	0.63 ($\sigma = 0.10$)

Table 2: Fleiss’ kappa and Krippendorff’s alpha agreement scores among coders for each refinement iteration.

3.3.3 Step 3: Coding the remainder of the dataset. We then proceeded to coding the remainder of the data set, which consisted of 595 figures, 460 were extracted from the ACM proceedings and 135 from patents. Since the agreement score stabilised among coders, only one coder effectively took part in this step.

We used two metrics to evaluate the precision of this coder. The first metric was subjective and simply consisted in giving a confidence score from 1 to 5 after coding each figure in the dataset (average confidence score was of 4.3, $\sigma = 0.8$). The second metric was objective and consisted in computing a stability score. We duplicated 5% of the figures in the dataset (*i.e.* 29 figures) in order to control that the coding was consistent across the duplicate figures. The stability score computed for each figure with Krippendorff's alpha and Fleiss' kappa, was on average about 0.8 ($\sigma = 0.1$, $min = 0.5$, $max = 1.0$) for both metrics. Note that only one figure had a score of 0.5 and eight a perfect agreement score of 1.0 considering the average number of codes selected for the 29 figures is 16.19 codes.

3.3.4 Step 4: Recode the figures used to elaborate the taxonomy. Finally, the coder who coded the remaining of the dataset in the previous step made an additional pass on the 200 figures that were used during steps 1 and 2, in order to update their coding this time with the definitive codes of the taxonomy. This step is required in order to homogenise codes for all figures of the dataset and then be able to analyse the entire dataset rather than limit the analysis to the figures coded in step 3.

Note that during steps 3 and 4, respectively 18 and 4 figures were excluded as these figures were eventually not considered as representations of interactive scenarios. This subset corresponds to 2.8% of the dataset used for the taxonomy. In the end, our coded dataset is composed of 773 figures, each coded with a set of what and how codes.

4 ILLUSTRATION STRATEGIES OF INTERACTIVE SCENARIOS

Eventually, we gathered a dataset of 773 figures, each of them coded with a set of *what* and *how* codes. Our main goal is to find relationships between uses of codes and figures, *i.e.* finding relevant groups of figures using a same group of codes. Our hypothesis is that inside each identified group of figures, analysing the relationships between *what* and *how* codes would yield to the understanding of strategies used to represent interaction. We define a strategy as the relationship between a group of *what* and/or *how* codes in a set of figures. One assumption is that, the more a given group of codes is present in different figures, the more an underlying strategy is expected to be used to represent the concept illustrated. We did not find any appropriate or standard method to analyse this multidimensional problem, therefore we chose to find the strategies using an exploration-based approach with our own exploratory tool (described in section 5). During the data exploration, we identified three types of strategies: **Structural Strategies (SS)**, **Interaction Strategies (IS)** and **Patent Strategies (PS)**. **Structural Strategies** are strategies that can be applied to any figure, regardless of the type of interaction illustrated, as they are mostly composed of *how* codes groups. They provide guidelines on how to deal with the basics of figure conception. To find these strategies, we applied a bottom-up methodology. We successively selected *how* codes in our exploratory tool to observe the impact on other *how* codes and *what* codes without further treatments. We then analysed the resulting figures to understand tendencies described in the following sections. We identified five structural strategies: choosing a frame

structure (**SS1**), choosing a point of view (**SS2**), sub-framing (**SS3**), defining relationships between elements (**SS4**), and preprocessing photographs (**SS5**).

Interaction Strategies are strategies that apply only to specific interactive contexts. They are composed of groups of *how* codes and *what* codes and thus, provide guidelines about representing a specific component of the figure. To identify these strategies, we applied a top-down and grouping methodology. We successively selected *what* codes, and we recursively selected the most common *what* codes present in the current figure set until the formed group was relevant. Then, we observed the most present *how* codes in the resulting figure set to obtain the strategy. We identified five interaction strategies: defining the interactive space (**IS1**), highlighting interactive elements (**IS2**), representing frame dynamic (**IS3**), highlighting users' actions (**IS4**) and illustrating non-visual feedback (**IS5**).

Patent Strategies were more straightforward to identify. As seen in the related work, illustrations guidelines already exist to standardise figures in patents. We used the same bottom-up methodology as for the general strategies to identify the strategies used in respect to these standardisation rules. We also highlight examples of patent figures showing that the introduced general and specific strategies are used to illustrate interactive contexts in patents.

All these strategies are described hereafter using figures from other papers as examples. Please note while some example figures may be a portion of the corresponding original figures, no other modification was applied.

SS1: Choosing a frame structure

Frame structure represents the layout of frames in the figure *i.e.* their number and how they are arranged.

Number of frames

414 (69%) of the figures in the data set use multiple frames. The number of frames directly depends on the interactive context. While depicting an interactive system itself might require only one frame, illustrating interaction sequences (Figure 1(a)) or design spaces (Figures 1(b), 1(c) and 1(d)) uses multiple frames to show the main steps of the sequence or the different possibilities of the designed interaction. Among figures using multiple frames, 35% represent interaction sequences and 55% design spaces.

Organising frames

Frames organisation depends on the content representation, the number of frames, and in particular on the space available for the figure. Interaction sequence frames are often organised using a one-dimensional layout to facilitate the succession of frames (usually from left to right, see Figure 1(a)). Design spaces do not need sequencing and provide more freedom to use different layouts, typically a grid layout, such as in Figures 1(c) and 1(d). Interestingly, note that these grid-layouts can be "nested", such as in 1(c) where each frame of a 2×2 grid (identified using letters, see below) is composed of two different frames using frame juxtaposition (discussed in SS3).

Identifying frames

When creating a figure with multiple frames, it might be necessary to indicate the order of the frames to the reader. These identifiers

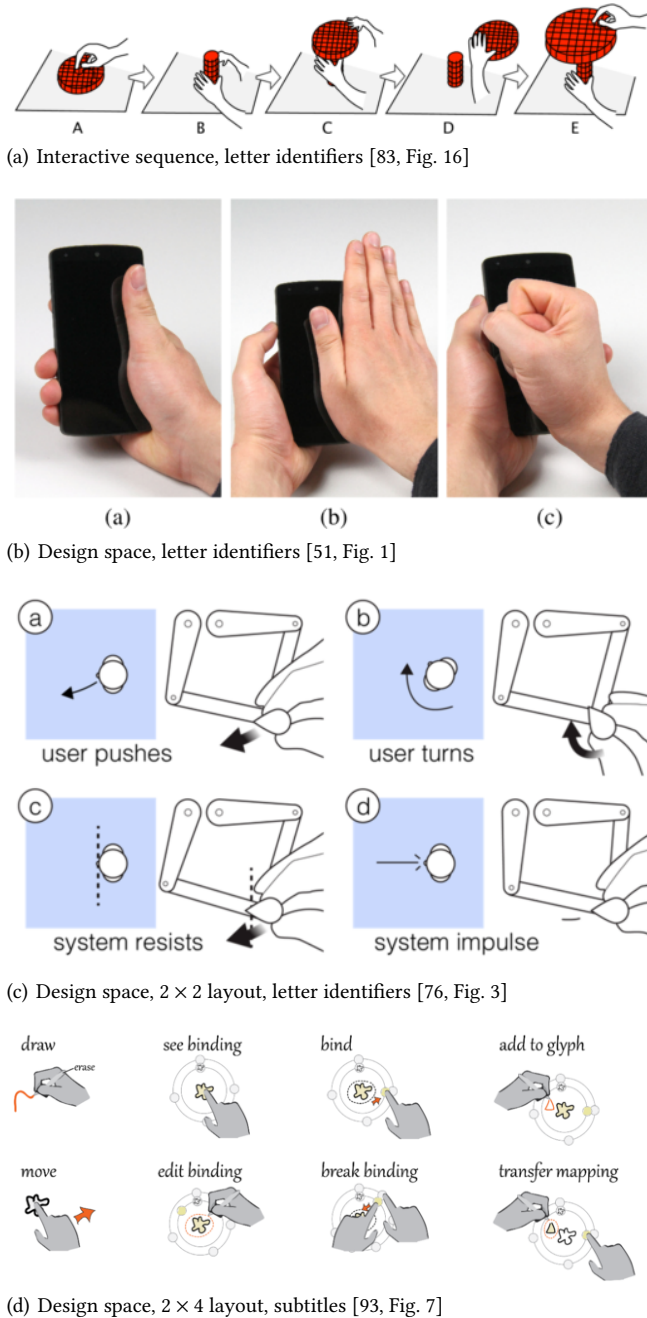


Figure 1: Examples of different frame structures

are generally displayed on a frame corner or border. To order frames such as in interaction sequences, letters or numbers are generally used to identify each step of the sequence (see Figure 1(a)). Letters and numbers are also used along with subtitles as frame identifiers

when ordering is not involved, for instance, in design spaces. For example, Figure 1(b) only uses letters, Figure 1(d) subtitles and Figure 1(c) both. As both letters and numbers are used for ordering and non-ordering purpose, the reader has to understand the context to guess the meaning of the identifiers. In that case, transition arrows between successive frames can be used as an extra indicator to reinforce the image sequence and guide the reading of the figure (Figure 1(a)). In the dataset 44% of figures used letters, 13% subtitles, 5% transition arrows and 2% numbers.

SS2: Choosing a point of view

Each figure frame uses a specific point of view, *i.e.* an observation point with a particular perspective on the represented interactive context. We identified four main points of view that depend on the interaction illustrated: 3rd person, 1st person, top view, and over-shoulder.

3rd person view

The most frequently used perspective, with 323 figures (54%), depicts a user interacting with a system, or the system itself, from a 3rd person's point of view. For example, Figure 2(e) shows a user wearing an HMD and Figure 2(f) illustrates the user posture in a context of desktop interaction.

1st person view

Another frequently used point of view shows the scene from the user perspective, using a 1st person view (170 figures, 28%). Depending on the context, the way this point of view is used may vary. Typically, in VR (24%) or AR (14%), the 1st person view is often a capture of what the user sees through the HMD with the corresponding user interface (see Figures 2(e) and 2(c)). For other contexts, using a 1st person perspective is mostly used to focus on user's hands or device such as in a touch interaction context (31% represent mobile devices). As an example, Figure 2(d) shows specific touch gestures to use on a smartphone and the hand posture.

Top view

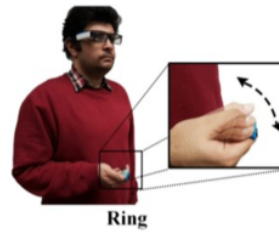
Representing the scene from a top view (101 figures, 17%) can be used to get a wide angle of a scene to observe and ease the readability of user interfaces by avoiding perspective. For instance, top view is mostly used in touch (54%) and tangible (20%) interactions as it allows to get a view on the user interface, user limbs such as hands or fingers and the devices used. As examples, Figure 2(a) represents the tangible objects and the interactive surface and Figure 2(b) illustrates specific gestures on a smartphone.

Over-shoulder view

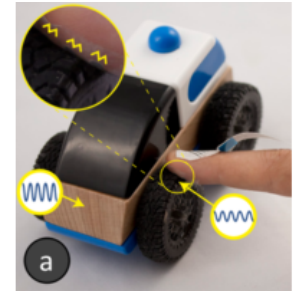
Used in 55 figures (9%), this point of view aims to illustrate the distance between the user and the system she is interacting with. While not specially used for distal interaction, 21% of figures concern distal pointing tasks. By positioning the point of view above the user's shoulder, it is possible to observe the UI and the direction pointed by the user, as illustrated in Figures 2(g) and 2(h) representing distal wall pointing and TV remote control.



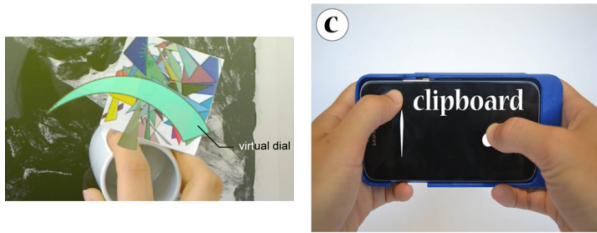
(a) Top view [41, Fig. 5] (b) Top view [30, Fig. 2]



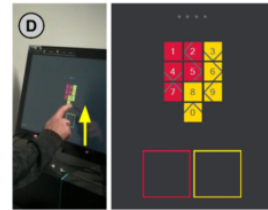
(a) Frame magnification [3, Fig. 1]



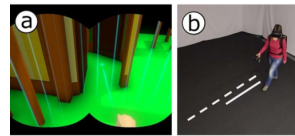
(b) Frame magnification [90, Fig. 5]



(c) 1st person view [56, Fig. 3] (d) 1st person view [86, Fig. 2]



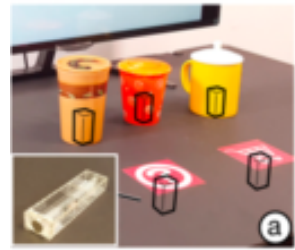
(c) Frame juxtaposition [45, Fig. 1] (d) Frame juxtaposition [60, Fig. 12]



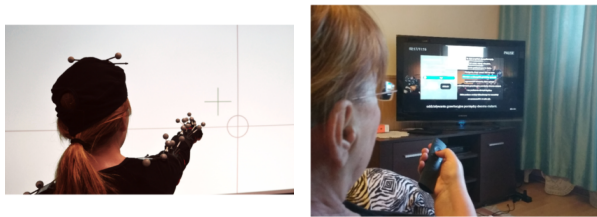
(e) 3rd person view [16, Fig. 5] (f) 3rd person view [92, Fig. 1]



(e) Picture-in-picture [10, Fig. 3]



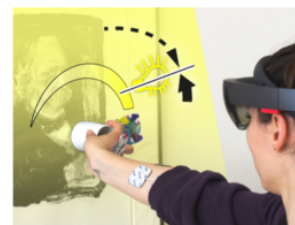
(f) Picture-in-picture [53, Fig. 4]



(g) Over-shoulder view [63, Fig. 5] (h) Over-shoulder view [80, Fig. 6]



(g) Frame blending [94, Fig. 11]



(h) Frame blending [56, Fig. 6]

Figure 2: Examples of different points of view

Figure 3: Examples of different sub-framing strategies

SS3: Sub-framing

Given a frame structure for a figure and a point of view for each frame, different strategies can be used to combine visual elements using sub-framing.

Frame magnification

Frame magnification consists of embedding subframes in the main frame to represent a zoomed portion of the figure. These subframes are used to provide focus and context at the same time, by adding sub-level information that cannot be represented at the main figure level while keeping the same perspective. The zoomed area

is typically enclosed and linked to the subframes. We counted 24 figures (4% of the dataset) using frame magnification. In Figure 3(a), the lens provides details about a specific hand gesture with a stroboscopic effect and a motion arrow that would be hard to notice at the figure level. In Figure 3(b), the lens gives a detailed representation of haptic feedback.

Frame juxtaposition

Subframes can also be juxtaposed and separated by a line or white space, vertically or horizontally. This strategy is often used to represent different perspectives on the same figure. 80 figures (13% of the dataset) use frame juxtaposition, from which 25% correspond to a VR interaction context. Figures 3(c) and 3(d) are two examples of juxtaposed subframes use, where one subframe represents the user’s perspective and the other the user interface.

Picture-in-picture

With Picture-in-picture (PiP) one frame is embedded into another. In contrast to frame magnification, the two subframes do not represent the same image at different zoom level. Picture-in-picture was used in 38 figures (6%) in our dataset. Figure 3(e) and 3(f) show two use cases of PiP, such as illustrating different points of view in the same figure (from different users for example) or showing details about a created device used in a system.

Frame blending

Finally, frames can be blended, by overlaying subframes (often two) on each other using transparency. This strategy is mostly used to overlay virtual/digital content on the real world representation. Frame blending was used in 47 figures (7%) of our dataset, from which 67% illustrate AR interaction. For example, Figure 3(g) overlays with transparency the interactive space boundaries for the user in VR. Figure 3(h) overlays an AR interaction user interface.

SS4: Defining relationships between elements

These specific strategies gather techniques used to group or link elements inside and outside the figures such as by associating a label or a definition to an element, or grouping multiple elements across one or multiple frames. Three strategies are used for that purpose: colour, shape and identifiers grouping.

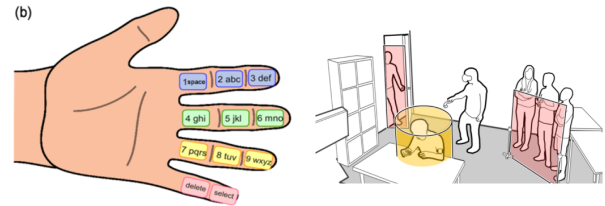
Grouping using colours

Colour grouping (used in 73 figures, 12% of our dataset) consists of grouping elements for a figure using the same colour for filling, strokes, or even text. For example, interactive areas in Figure 4(a) are displayed using different colours, with areas corresponding to the same finger displayed with the same colour. Figure 4(b) groups the global interactive area borders in red and the personal area in orange. Figure 4(c) uses different colours, green for interactive objects and blue for the interactive area. Finally, Figure 4(d) uses colour grouping to link figure elements to a legend.

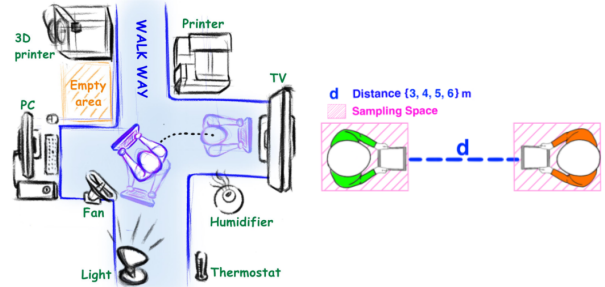
Grouping using lines and arrows

Simple geometric shapes can also be used to group elements. Typically, arrows can be used to group an object of the system with a descriptive element (typically a label), while lines can be used to connect multiple objects of a figure. For instance, Figure 4(e) uses yellow arrows to add extra information to describe the vibrations generated by haptic elements. Figure 4(f) on the other hand uses

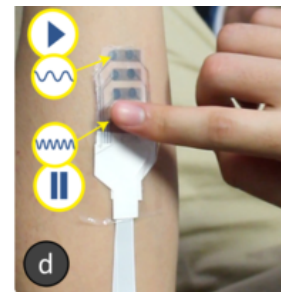
plain lines with a label to link real objects in the photography to the corresponding virtual objects on the display. 79 figures (12%) use arrows and lines grouping, from which 71 figures use additional text annotation for grouping.



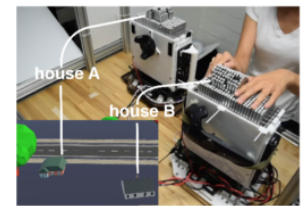
(a) Area colour grouping [91, Fig. 1] (b) Area colour grouping [94, Fig. 1]



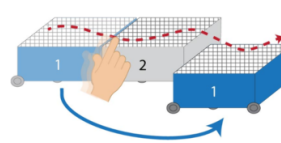
(c) Objects colour grouping [39, Fig. 8] (d) Legend colour grouping [40, Fig. 6]



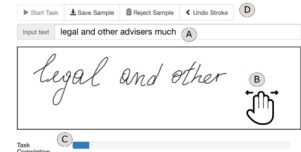
(e) Arrow grouping [90, Fig. 5]



(f) Line grouping [79, Fig. 9]



(g) Number grouping [79, Fig. 7]



(h) Letter grouping [2, Fig. 14]

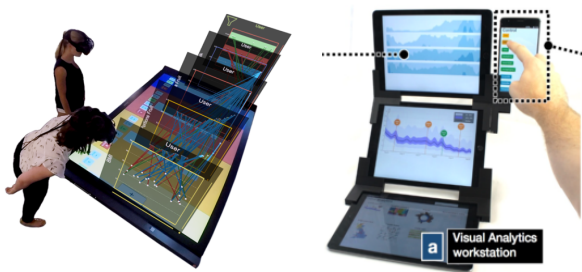
Figure 4: Examples of strategies used to define relationships between elements

Grouping using identifiers

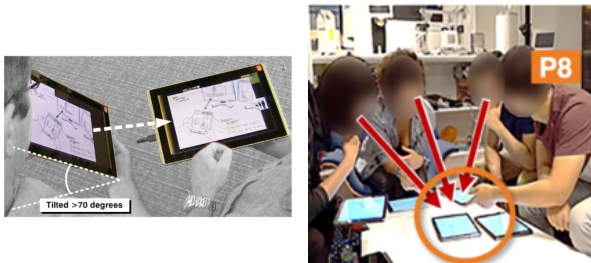
Finally, identifiers such as letters or numbers are also used to provide structure. These identifiers are often placed on the object itself or next to it. Figure 4(g) uses the number 1 twice to show that the blue transparent and opaque blue parallelepipeds are the same object that is actually moved. Figure 4(h) uses letter identifiers A, B, C and D to link specific parts of the figure to extra information outside the figure such as definitions in the caption (not included). We found 31 figures (5%) using identifiers for grouping in our dataset.

SS5: Preprocessing photographs

Two strategies, specific to photographs, consist of pre-processing the raw medium (i.e., the source photo) in order to highlight specific elements, or anonymize users.



(a) Background removed [12, Fig. 6] (b) Background removed [57, Fig. 18]



(c) Background greyed out [58, (d) Anonymisation blur [11, Fig. 5] Fig. 4]



(e) Anonymisation filling [23, Fig. 1]

Figure 5: Examples of photograph preprocessing strategies

Background removal

We found 12 figures (2%) that removed the background and two figures greying out the background in our dataset among the 387 figures using photographs. While uncommon, it corresponds to a strategy which consists in removing or blurring the background of a photograph in order to make the main elements such as the users and the system stand out, while at the same time remove unnecessary details that might distract the viewer's attention or result in visual clutter. Figures 5(a) and 5(b) are good examples of users and devices detouring. Figure 5(c) [58] shows a greyed background highlighting the tablets.

Anonymisation blurring or filling

Two strategies can be used to anonymize users. The simplest one consists in blurring portions of the figures where users faces appear (see Figure 5(d)). We only found 9 figures using such anonymisation technique in our dataset. Another type of anonymisation (not present in our dataset) might also be used, such as filling all the areas of the figures showing a person in order to anonymize their identity. Figure 5(e) [23] is a good example of such technique. Buxton [14] and Greenberg *et al.* [31] refer in this context to the technique of hybrid sketching: a part of the original photograph is kept, while other parts are sketched over (in this case here in order to anonymize). Besides anonymisation, hybrid sketching can be used more broadly for any interaction scenarios where it is important to provide rich information about the context of the scene (through the photograph), and overlaying these photographs with the sketched elements illustrating an interaction technique, device, or interface.

IS1: Defining the interactive space

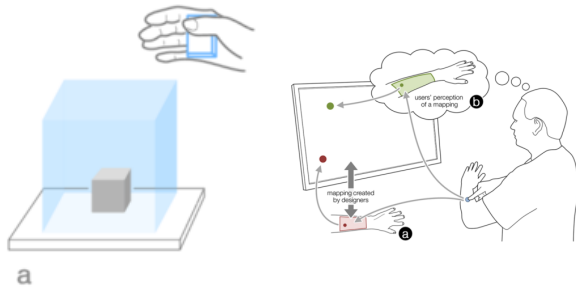
Highlighting the physical dimensions of an interaction space (e.g., the two-dimensional area or a three-dimensional volume) is useful to show a constrained space or a specific environment a person can interact.

Colouring the interactive space

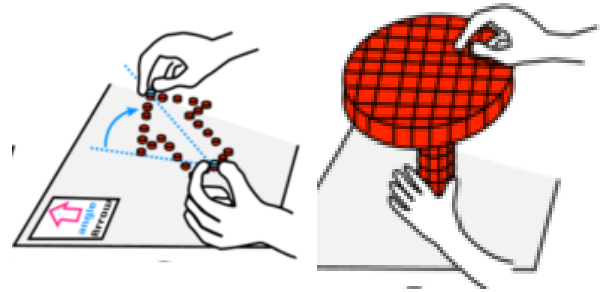
Applying a specific colour to a portion of the figure is a strategy that is often used to highlight a 2D interactive space. Figures 6(b) and 6(c) provide good examples of coloured interactive spaces. The former shows a constrained on-skin interactive touch surface limited to the arms of the user. The latter shows the area of possible user displacements in a distal interactive context. Highlighting 3D spaces with colour can also be done using a geometrical 3D shape enclosing these physical spaces. Figure 6(a) demonstrates this technique where a semi-transparent blue cube is used for the 3D volume a person can interact with the system.

Highlighting the borders of a space

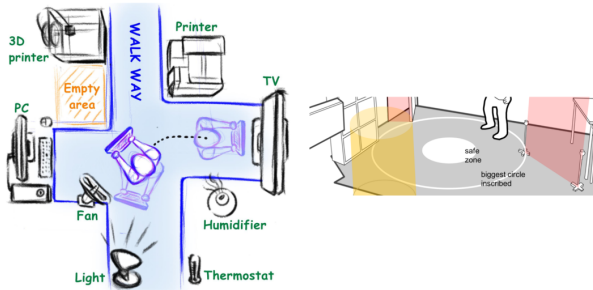
Instead of directly highlighting the interaction space, some figures use space delimiters to provide spatial structure. A classic approach is to draw lines on the figure as space delimiters. Figures 6(d) and 6(e) are good examples of the use of lines: while both figures are delimiting VR spaces, the first one is using plain white circles, and the second one dashed lines. Figure 6(f) instead uses transparent coloured panels embedded in the figure as borders for each individual interaction space in the VR environment.



(a) Colouring 3D space [84, Fig. 9] (b) Colouring 2D space [6, Fig. 1]



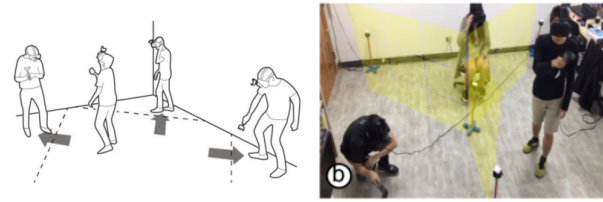
(a) Colour focus [82, Fig. 13] (b) Colour focus [83, Fig. 16]



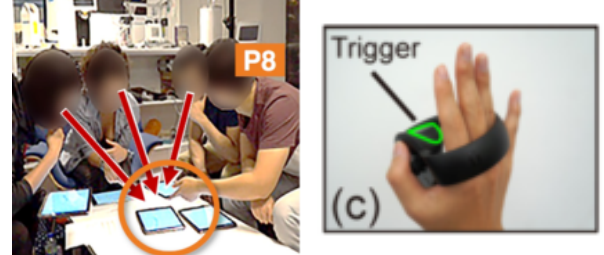
(c) Colouring 2D space [39, Fig. 8] (d) Border circles [94, Fig. 10]



(c) Transparency focus [33, Fig. 4] (d) Basic shape enclosing [57, Fig. 18]



(e) Border dashed lines [61, Fig. 4] (f) Transparent borders [94, Fig. 15]



(e) Basic shape enclosing [11, Fig. 5] (f) Exact contour enclosing [52, Fig. 7]

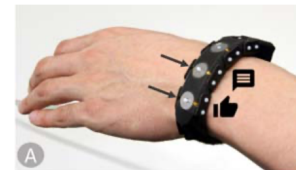
Figure 6: Examples of figures highlighting the interactive space

IS2: Highlighting interactive elements

These strategies highlight interactive elements through colour and transparency, elements enclosing, or pointing using arrows.

Using colour and transparency

Colour is often used to highlight specific elements, a strategy observed in 117 (19%) figures of our dataset. For example, Figure 7(a) highlights the tangible objects in red, while lines and arrows used for illustrating the current manipulation angle and direction are displayed in blue. In some cases, only a single element is coloured while the remainder of the figure is represented in grey scale (e.g. Figure 7(b)). Transparency can also be used to provide focus on an interactive element. For example, to emphasise the hand/gestural interaction, Figure 7(c) uses a transparency layer only for the user’s hands.



(g) Arrow pointing [67, Fig. 5]

Figure 7: Examples of figures highlighting interactive elements

Enclosing

Another technique to emphasise specific elements in a figure is to use enclosing. Enclosing consists in surrounding elements with lines so that the focus is put on them, often using different line styles (e.g. colour, plain or dashed lines). We distinguish two types of enclosing: basic shape enclosing (40 figures, 6%) and exact contour enclosing (15 figures, 2%). Basic shape enclosing consists in using a simple geometric shape to enclose the emphasised element. For example, Figure 7(d) uses a dashed-line rectangle to emphasise the smartphone and Figure 7(e) uses an orange-coloured circle to emphasise the tablet. Enclosing can also follow the exact external contour of the focused element. For example, Figure 7(f) highlights the controller trigger by using a green-coloured contour.

Arrow pointing

Although using arrows pointing at an object is a simple technique to emphasise an element, surprisingly we did not find an example of this technique in our dataset. Figure 7(g) (extracted from paper [67]) is a good example of applying this technique, where arrows highlight magnets attached to the wearable device.

IS3: Representing dynamic changes over time in a single frame

While dynamic behaviours in a figure can be represented using multiple frames to show the different steps of a sequence or motion, there are also a few techniques to illustrate dynamic into one frame. In this section we briefly describe such techniques, and refer to McAweeney *et al.* for a specific taxonomy on the representation of gestures [64].

Stroboscopic effect

Stroboscopic effects consist of drawing previous states/positions of a moving element (e.g. controller, device, body part) using different colours, transparency or line styles. Figure 8(a) shows a drawing-based stroboscopic effect illustrating a thumb scroll on a touchscreen. Figure 8(b) illustrates a photography-based stroboscopic effect of a person’s moving body.

Lines and arrows

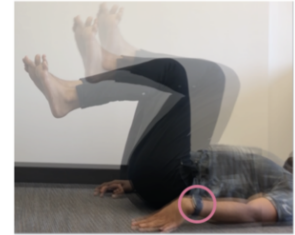
Lines and arrows are simple elements that can be used to show dynamic actions. Typically, lines can be used to describe motion paths (e.g. of a device, a body part or a cursor), as in Figure 8(e) where a motion path is represented by a transparent pink line on the photography. Arrows are used for directions and trajectories. Figure 8(c) shows two possible rotations for the finger to manipulate the cube using two blue arrows. Figure 8(d) shows the trajectory of the user on the room’s floor.

Onomatopoeias

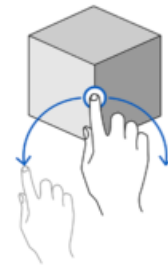
Also described as visual metaphors for what cannot be seen (p. 128, [65]), *onomatopoeias* are a collection of small details added to a figure, directly inspired from techniques used in comics, to give the impression of dynamic actions. We distinguish two types: textual and visual onomatopoeias. For example, small strokes can be used behind an object to show a certain velocity (Figure 8(g)) or arranged in a circle around a mouse cursor to illustrate a mouse click (Figure 8(f)). Figure 8(h) uses text onomatopoeia to illustrate smartwatch vibrations.



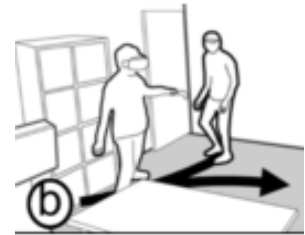
(a) Drawing-based stroboscopic effect [21, Fig. 1]



(b) Photography-based stroboscopic effect [46, Fig. 3]



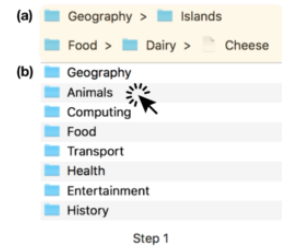
(c) Direction arrows [64, Fig. 9]



(d) Trajectory [94, Fig. 12]



(e) Motion path [40, Fig. 10]



(f) Visual onomatopoeia: click effect [55, Fig. 1]



(g) Visual onomatopoeia: speed effect [56, Fig. 1]



(h) Textual onomatopoeia: vibration effect [70, Fig. 1]

Figure 8: Examples of strategies used to represent dynamic in a single frame

IS4: Highlighting users' actions

We identified several strategies aiming at emphasising user interaction with a system.

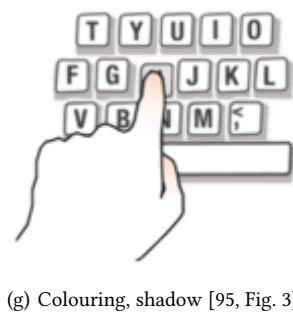
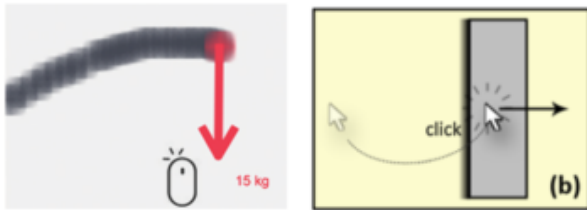
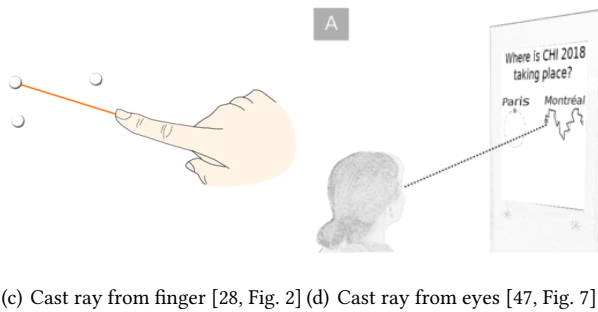
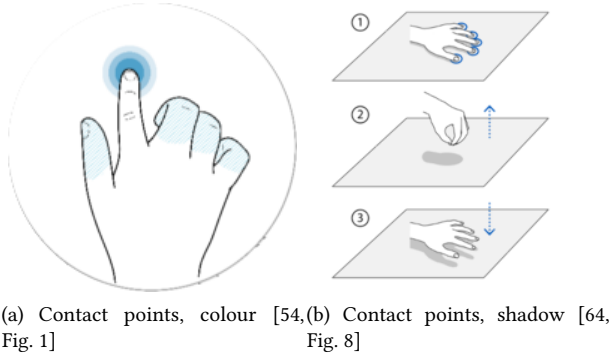


Figure 9: Examples of strategies used to highlight users' actions

Drawing contact shapes

A strategy to represent physical contact between two elements is to highlight the contact area by drawing a shape (typically a disk) as an intermediate layer between the elements. This technique is commonly used to represent a touch contact between a finger and a touch-sensitive surface (25 out of the 42 figures using contact shapes illustrate touch interaction). As examples, figures 9(a) and 9(b) use blue circles to highlight contact points.

Showing raycast direction

Lines or arrows can be drawn to show a ray cast projection (visible or not) from the user (using controllers, hands, fingers or eyes) to the system. Such projection lines are usually displayed using a specific colour or style. For instance, Figure 9(c) shows an orange-coloured ray cast from the user's finger. Similarly, Figure 9(d) shows a dashed line representing the ray cast from the user's eyes in a context of gaze interaction.

Drop shadow effects

Using different types of drop shadows can help to illustrate physical distance. For example, Figure 9(b) does not show any shadow of the hand when it touches the surface, but displays a shadow when the hand is in mid-air. Conversely, Figure 9(g) removes the shadow of the pressed key while changing its colour from white to light grey.

Showing UI cursor

We can explicitly add a system cursor to better understand the interaction, for example when using mouse input. Figure 9(e) displays a mouse icon embedded in the figure and Figure 9(f) shows the mouse cursor. Furthermore, both figures use a *clicked effect*, i.e. small strokes arranged in a circle around the icon, to represent a physical press of the mouse button.

Colouring interactive elements

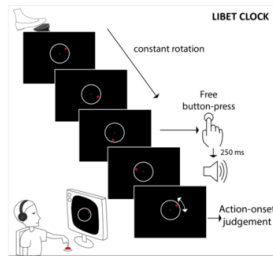
A different colour for the interactive elements may be used to highlight which elements of the system or which parts of the users can be used to interact (e.g. remotes, controllers, fingers). For example, Figure 9(a) uses a white finger (versus light blue) to show the finger on the touch surface. Figure 9(g) uses light orange to show which finger presses the keyboard H key.

IS5: Illustrating non-visual feedback

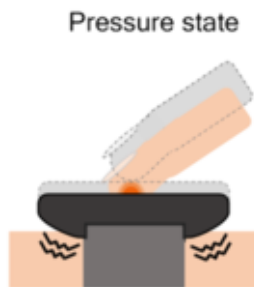
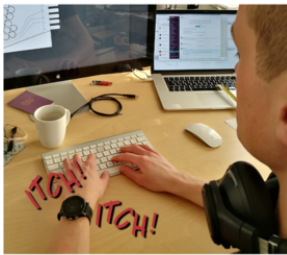
Non-visual feedback such as audio or haptic feedback can be challenging to represent on a static illustration. In this section, we explain a few strategies used to overcome this difficulty by adding additional information on the figures.

Audio feedback

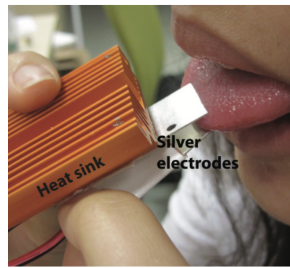
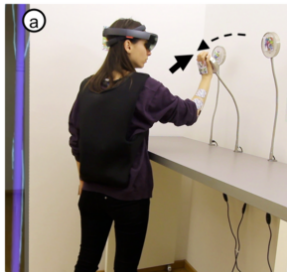
21 figures (3% of our dataset) indirectly illustrate audio feedback by showing the audio-device used (e.g. speakers or earphones) that let the reader infer there is an audio response provided by the interactive system. For instance, Figure 10(a) shows a user playing a racing game with headphones on his head suggesting the use of sound feedback. Another strategy is to add an icon representing sound in the figure. This is the case for Figure 10(b) that illustrates a timeline of events, with a transition arrow going from a button



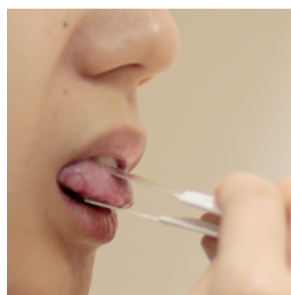
(a) Suggested audio feedback [81, (b) Speaker icon [59, Fig. 5] Fig. 1]



(c) Vibration text [70, Fig. 1] (d) Vibration waves effect [77, Fig. 1]



(e) Arrow force feedback [56, (f) Taste feedback close-up [75, Fig. 12] Fig. 4]



(g) Taste feedback close-up [74, Fig. 2]

Figure 10: Examples of strategies used to illustrate non-visual feedback

press to a sound icon meaning once the button is pressed, a sound is emitted by the system.

Haptic feedback

Unlike audio feedback, haptic feedback such as vibration or force feedback is illustrated using dynamic illustration strategies. We first observed the use of onomatopoeias, mentioned earlier in the paper. Figure 10(c) uses the *ITCH* text twice with style to illustrate smartwatch vibration, while Figure 10(d) uses an effect of waves. Arrows are also used, specifically for force feedback interactions, where different line styles are used to represent the different forces applied by the system (Figure 10(e)).

Taste feedback

While our dataset does not contain figures representing interaction involving the sense of taste, we had to analyse papers about multi-sensory interaction such as [74, 75] in order to extract a strategy, which consists in using a close-up 3rd person perspective of the user's mouth with the interactive device in the mouth to show electrodes in contact with user's tongue and suggest taste feedback (figures 10(f) and 10(g)).

Patents strategies

Aspect

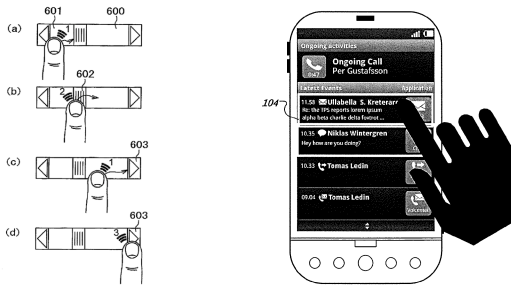
Patent figures are all done in a black and white or greyscale colour scheme, and primarily by using line drawings. Some patent figures embed photographs where colours have been converted to greyscale (see Figure 11(b)). Drawing style ranges from realistic (Figure 11(f)) to schematic (Figure 11(h)). User interfaces and wire frames are mostly drawn in the image plane without perspective such as Figure 11(e).

Identifiers

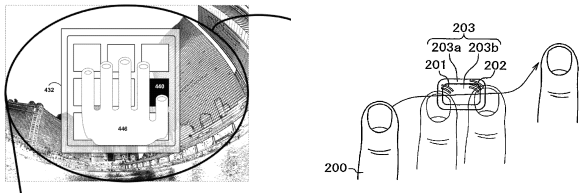
Patent figures all use a specific identifier to link elements to definitions outside the figure. Related to guidelines, these identifiers are always composed of a number and an arrow/line pointing at an element.

Use of interactive strategies

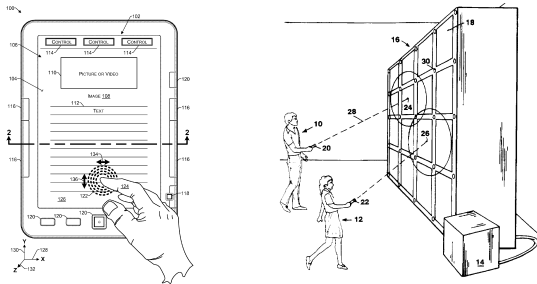
Patent figures also use strategies similar to the ones we identified in the figures extracted from academic conference proceedings. Different points of view are used: top (Fig. 11(a)), 1st person (Fig. 11(d) and 11(e)) or 3rd (Fig. 11(f)). Poupyrev *et al.* illustrated an interaction sequence using two strategies on their patent [72]: multiple frames (Fig. 11(a)) and stroboscopic effect (Fig. 11(d)). Fig. 11(c) uses frame blending to represent the AR view of the system and the background. Interactive areas can also be highlighted using a darker grey colour 11(g). Another technique consists in drawing a field of rays from the camera to define the interactive area, as shown in Figure 11(h). As identified before, interaction between users and systems are also represented using contact points for touch interaction (Fig. 11(e)) or rays for distal interaction (Fig. 11(f)).



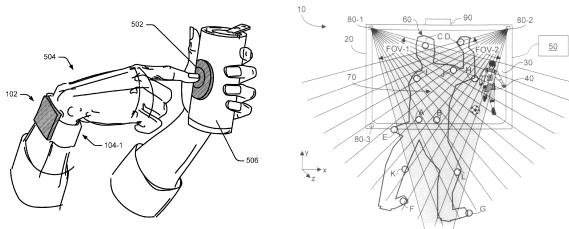
(a) Multi-frame [72, Fig. 10] (b) Image preprocessing [87, Fig. 1]



(c) Frame blending [69, Fig. 4] (d) Stroboscopic effect [72, Fig. 3]



(e) Touch contact points [13, Fig. 1] (f) Cast rays [42, Fig. 1]



(g) Element coloured [71, Fig. 5] (h) Interactive area [73, Fig. 1A]

Figure 11: Examples of figures used in patents to illustrate interaction

5 OPEN-SOURCE TOOLS FOR TAXONOMY CODING, VISUALISATION AND EXPLORATION

We developed three tools that facilitated the task of creating our taxonomy and identifying visual strategies. These tools were critical for allowing an effective coding and review/analysis of the large dataset of visuals. In particular, we designed and implemented two dedicated tools supporting coding (Figure 12), and an additional tool for exploring the final taxonomy and identifying strategies. We provide these tools as open-source, to support other researchers in our community when creating taxonomies in the future. All three tools are released as open-source software, available at <https://ns.inria.fr/loki/IllustrationTaxonomy/>.

5.1 Coding tool

First, we built a coding application in PyQt (Figure 12(a)). It displays one figure at a time and its corresponding caption to help coders understand the displayed figure. On the right side of the app, a tree widget displays the code hierarchy. Hovering a code with the mouse displays a tooltip containing the code description. For each leaf of the code tree, a checkbox can be used to select or deselect this code for the given figure. We also added the possibility to add, remove, and rename codes in the hierarchy, which was very useful for the first iterations when changes were needed. Keyboard shortcuts can be used to rapidly interact with the application and optimise the workflow.

Key categories of the taxonomy are highlighted with a red background in the tree widget when none of their children codes were selected, and the background changes to green when at least one child code is selected. A global indicator in the top right corner changes from red to green when all key categories are green. The key categories colour system helps coders to quickly identify when the tagging of a figure is completed, *i.e.* when at least one code in each key category was selected.

Finally, a text comment area and two 5-stars rating widgets can be used by coders to give feedback on their coding confidence and the overall subjective rating of the figure.

5.2 Visualisation tool

Second, we created an on-line visualisation tool using the D3 toolkit [9]. This tool, provided as supplementary material, proposes two types of visualisations for the taxonomy (Figure 12(b)): a simple expanding tree view (allowing clicking on each node of the coding hierarchy), and a treemap view (allowing observing examples of figures for each code of our taxonomy). Both visualisation tools allow reading each code description and were automatically updated upon each coding iteration. These on-line tools helped coders to become more familiar with the different codes of the taxonomy, and were also used as visual support for the refinement discussions.

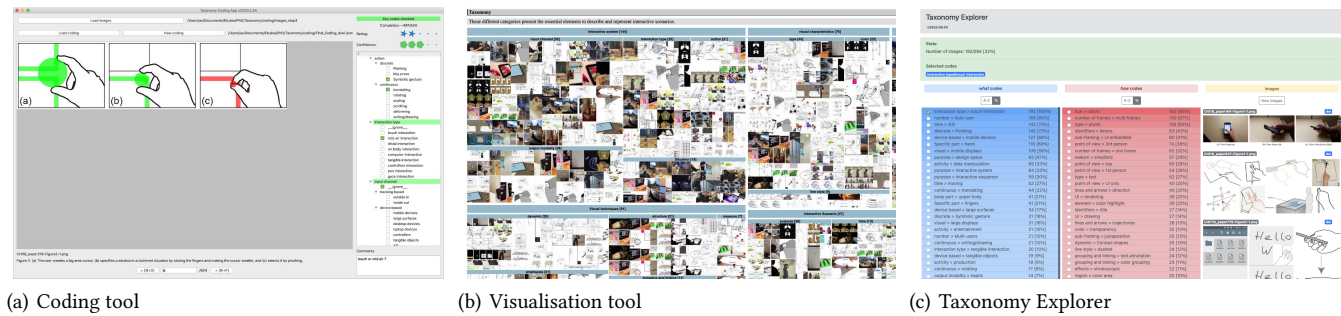


Figure 12: (a) The coding tool was used to tag the figures. It displays the figure with its caption, two 5-stars rating widgets to enter coding confidence and figure score, the hierarchy of codes as a tree widget and a comment area. (b) The visualisation tool was used to navigate through the hierarchy of codes, see codes and categories, their definition and figure examples. (c) The Taxonomy Explorer is a web application to navigate through the coded data. Selecting *what* code and *how* code items (respectively in the blue and red lists) apply a filter on the data set to get the number of figures with these codes. Both lists can be sorted alphabetically or by frequency, which are updated in real time upon the current codes selection. The right column shows examples of figures corresponding to the selection.

5.3 Exploration tool

Third, we built a visual exploration software tool, *Taxonomy Explorer* (see Figure 12(c)), that can be used to browse the codes used in the taxonomy, sort and filter them by frequency of use, and display figures associated with the codes that are selected. The main idea of this tool is to be able to select specific codes and observe the impact of the selection on the other codes and figures. This tool is composed of four main parts: a green horizontal banner, and three vertical columns (one blue, one red and one displaying figures). The blue and red columns respectively display the list of *what* and *how* codes of our taxonomy. Each row of these columns is an HTML structure containing: a checkbox to (de)select the code, the code name (concatenated with its parent category name), and numbers indicating occurrences and percentage of the code in the set of figures corresponding to the currently selected codes. When (de)selecting a code, the set of figures corresponding to selected codes (*i.e.* the intersection) is automatically retrieved, and the interface updated to reflect the code (de)selection. The rows in the blue and red columns also have background colours of different brightness, brighter corresponding to a higher percentage of the code in the corresponding set of figures. If no code is selected, all figures are included in the selection. In order to support different code exploration strategies, both lists can be dynamically sorted alphabetically or by frequency. The green horizontal banner shows the list of currently selected codes. Clicking on one of them removes it from the current selection and un-checks the corresponding checkbox in the blue or red column. Finally, the last column displays a dynamic scrollable view with examples of figures that have been tagged with currently selected codes. As mentioned in the previous sections, this tool was used to observe global tendencies of codes within the dataset, as well as identify illustration strategies.

6 DISCUSSION AND CONCLUSION

6.1 Reflection on use of taxonomy and tools

In this work, we developed a taxonomy of figures illustrating interactive scenarios and identified strategies that are used (and can

be reused) when producing such figures. The taxonomy was elaborated in a two-step process which consisted in having a group of coders producing codes from scratch, and iterating to refine these codes and their corresponding definitions. This approach was inherently driven by our dataset that consisted of figures illustrating interactive scenarios extracted from the proceedings of four major HCI conferences. While not necessarily complete, this dataset made it possible to expand beyond what had already been known about figures of interactive scenarios, and allowed us to create a comprehensive taxonomy of the representation of interaction in static visual representations. As illustrated on Table 1, our taxonomy integrates previously published categorisations of the representation of interaction, and extends beyond these with categories that were not yet or only partially considered by previous works.

The complete coding of this dataset then provides interesting insights on the illustration of interactive scenarios, as shown by the summary statistics that can be found in section A. These illustrations employ diverse layouts, structures, and points of view, showing that there is not one single approach to illustrate interaction and that guidelines and strategies are helpful depending on what an author might want to illustrate. The exploration of this dataset – supported with our (open-source) software tools – allowed us to identify Structural and Interaction strategies. We hope that our generalised strategies will function as guidelines and inspiration for HCI researchers and students to produce the best possible visual representation of their interactive scenario.

Our taxonomy and classification can likely be extended in the future, and additional strategies could be extracted by repeating our methodology either with a different dataset or with different goals in mind. Indeed, we provide our tools and complete dataset for this reason as open-source, to further allow the creative exploration of our taxonomy (or different taxonomies in the future). First, our *visualisation tool* allows the exploration of our taxonomy, to navigate our codes hierarchy, read codes descriptions and observe examples of figures for each code. Second, our *Exploration tool* can

be used to explore our dataset of coded figures and observe tendencies or discover new strategies. It is indeed important to note that our focus was on the illustration of interaction on static visual representations, but our taxonomy could be further extended to other fields, such as instruction manuals where specific strategies are used to help the user understand the different operations [1].

6.2 Reflection on methodology

We combined a grounded theory approach [50] which consists in building a theory through gathering and analysis of data, in our case, the manual analysis of the figures from our dataset. Part of the coding process used to build the taxonomy has been performed by several coders, until we observed a stable agreement score between them. After that, only one coder completed the process and then manually analysed the coded figures to identify strategies.

6.2.1 Measuring agreement between coders. To the best of our knowledge, there is no standard methodology for computing an agreement score in a coding process. Indeed, commonly used metrics [26, 48] for measuring inter-rater agreement were designed for categorisation tasks where an object can be classified in one and only one category, while in our case each figure could be assigned with multiple codes. To overcome this issue, we used a common adaptation of the Fleiss' kappa and Krippendorff's alpha to a coding context by computing an agreement score for each object (in our case each figure) instead of the whole set. In that respect, we considered within a figure each code as an object to assign to one of the non-overlapping categories: *Selected* or *Not-selected*. Eventually, overall agreement score was computed as the average of the agreement score of each figure. We decided to report agreement scores in spite of this adaptation to follow recommendations during coding procedure. However, this method remains an adaptation of what they were designed for, and as such, need to be carefully considered when used in other contexts.

6.2.2 Manual exploration of coded figures. We decided to implement the Exploration tool to support the manual exploration of the coded figures in order to extract strategies. Before that, we considered to explore the coded dataset using PCA or clustering methods such as K-means, but realised these techniques were not adapted. First, clustering methods usually partition all the objects of a dataset into a predetermined number of clusters. While fuzzy clustering [24] such as fuzzy c-means [8] could overcome this problem, algorithmic methods still would be unadapted because of the diversity of our dataset, both in terms of the number of codes and diversity of figures. Indeed, these algorithms are very sensitive to outliers, whereas our dataset has quite a few outliers because of underrepresented interaction types (e.g. VR was much more represented than speech input). Cleaning data to remove outliers or balanced weight between codes might be a solution, but would yield in a problem of finding cleaning criteria and eventually losing too much information. More importantly, we were also interested in analysing outliers and aimed to find strategies commonly used for a majority of figures, without missing relevant strategies sparsely used only because the interactive scenario depicted in the figure is rare. For instance, codes *foot body part*, *speech input*, *anonymisation blur*, *grey out* or *blurred background* and more than 10 others are used fewer than ten times in the whole dataset, and yet, most of

them are part of strategies we identified, presented in section 4. In the end, we believe that the methodology used was sound. It relies on a systematic coding of figures with a software-supported manual exploration of the coded dataset. The exploration of the dataset by a single coder already resulted in the extraction of several strategies of different types used to illustrate interaction.

6.3 Limitations of the study

We focused on identifying higher-level strategies for creating visual representations for interaction scenarios, ignoring lower-level strategies such as how visual variables (symbols, line styles, etc.) could be used. We decided to focus on the former as significant research has already been conducted on the latter [15] to the point that education books can be found in order to better understand how to adequately use these visual variables [89]. At the same time, several strategies (e.g. SS3 or SS4) would benefit from a more thorough understanding of semiology of graphics [7] or gestalt laws to optimise their implementation. While all the strategies reported in this paper can be seen as guidelines, this work does not provide any guidance regarding "good" or "bad" strategies to employ. On the one hand, we do consider all strategies reported in this paper as "good", which is why we believe they could be useful for the future creation of figures. On the other hand, we acknowledge that some strategies may be better than others and that distinction would require further work to be studied.

Another limitation of our work was that we did focus on figures illustrating interactive scenarios on tool/system/prototype use, ignoring other figures that can frequently be used in HCI papers such as data charts. Once again, we decided to focus on a largely under-explored area rather than focus on types of figures that have already been intensively investigated [35, 78, 89].

We also limited our investigation to the case of *static* figures, which remains the main visual medium to describe interactive scenarios in research papers. As such, we deliberately decided to ignore dynamic [32, 43, 44] and interactive figures [62] even though some can be found in recent HCI appears [29, 32], because they are not natively supported by most formats used for communicating about interactive scenarios (typically PDF research papers), and even when used, they require an equivalent static figure [32, 62].

6.4 Future work

In this work, we extracted structural characteristics and interactive strategies to illustrate interactive scenarios, readily used by the HCI community when producing their own figures. However, this still requires to implement these strategies by hand into our own figures. Recent systems were developed to support the production of illustrative figures, such as DemoDraw [20] or Esquisse [4]. A possible direction for future work would be to incorporate the strategies we identified as templates into these – or similar – tools, to simplify the reuse of such strategies.

A second direction for future work would be to evaluate the efficiency of the figures used in our dataset. Indeed, we can assume a certain quality level of all our figures given that they were extracted from research papers published in top tier HCI conferences, but is also likely that all figures are not of equivalent quality and that some of them convey more efficiently their meaning than others,

hence, that some of the identified strategies in this work could be more efficient than others. Future work could investigate this through user studies where participants produce several instances of an illustrative figure (in a similar way McAweeney *et al.* did for gesture representations [64]), using different strategies and comparing their overall performances through subjective feedback or ask participants to rate and describe what they understand for a given interaction scenario illustrated with different strategies.

REFERENCES

- [1] Maneesh Agrawala, Doantam Phan, Julie Heiser, John Haymaker, Jeff Klingner, Pat Hanrahan, and Barbara Tversky. 2003. Designing Effective Step-by-Step Assembly Instructions. In *ACM SIGGRAPH 2003 Papers* (San Diego, California) (*SIGGRAPH '03*). Association for Computing Machinery, New York, NY, USA, 828–837. <https://doi.org/10.1145/1201775.882352>
- [2] Emre Aksan, Fabrizio Pece, and Otmar Hilliges. 2018. DeepWriting. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. <https://doi.org/10.1145/3173574.3173779>
- [3] Fouad Alallah, Ali Neshati, Nima Sheibani, Yumiko Sakamoto, Andrea Bunt, Pourang Irani, and Khalad Hasan. 2018. Crowdsourcing vs Laboratory-Style Social Acceptability Studies?. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. <https://doi.org/10.1145/3173574.3173884>
- [4] Axel Antoine, Sylvain Malacria, Nicolai Marquardt, and Géry Casiez. 2019. Esquisse: Using 3D Models Staging to Facilitate the Creation of Vector-Based Trace Figures. In *Human-Computer Interaction - INTERACT 2019*, David Lamas, Fernando Loizides, Lennart Nacke, Helen Petrie, Marco Winckler, and Panayiotis Zaphiris (Eds.). Springer International Publishing, Cham, 496–516.
- [5] Mary Bellis. 2019. The Rules for Patents Drawings. Retrieved May 25th, 2020 from <https://www.thoughtco.com/rules-for-patent-drawings-1992228>
- [6] Joanna Bergstrom-Lehtovirta, Kasper Hornbæk, and Sebastian Boring. 2018. It's a Wrap. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. <https://doi.org/10.1145/3173574.3174138>
- [7] Jacques Bertin. 1983. *Semiology of graphics; diagrams networks maps*. Technical Report.
- [8] James C. Bezdek, Robert Ehrlich, and William Full. 1984. FCM: The fuzzy c-means clustering algorithm. *Computers and Geosciences* 10, 2 (1984), 191 – 203. [https://doi.org/10.1016/0098-3004\(84\)90020-7](https://doi.org/10.1016/0098-3004(84)90020-7)
- [9] M. Bostock, V. Ogievetsky, and J. Heer. 2011. D³ Data-Driven Documents. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (2011), 2301–2309.
- [10] Michael Brock, Aaron Quigley, and Per Ola Kristensson. 2018. Change Blindness in Proximity-Aware Mobile Interfaces. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) (*CHI '18*). Association for Computing Machinery, New York, NY, USA, 1–7. <https://doi.org/10.1145/3173574.3173617>
- [11] Frederik Brudy, Joshua Kevin Budiman, Steven Houben, and Nicolai Marquardt. 2018. Investigating the Role of an Overview Device in Multi-Device Collaboration. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. <https://doi.org/10.1145/3173574.3173874>
- [12] Simon Butscher, Sebastian Hubenschmid, Jens Müller, Johannes Fuchs, and Harald Reiterer. 2018. Clusters, Trends, and Outliers. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. <https://doi.org/10.1145/3173574.3173664>
- [13] David C. Buuck. 2013. Providing tactile output and interaction. <https://patentimages.storage.googleapis.com/af/b0/b3/241c27d8e77d55/US8593420.pdf> US Patent 8,593,420 B1.
- [14] Bill Buxton. 2007. *Sketching User Experiences: Getting the Design Right and the Right Design*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- [15] Sheelagh Carpendale. 2003. *Considering Visual Variables as a Basis for Information Visualisation*. Technical Report. University of Calgary. <https://doi.org/10.11575/PRISM/30495>
- [16] Hong-Yu Chang, Wen-Jie Tseng, Chia-En Tsai, Hsin-Yu Chen, Roshan Lalintha Peiris, and Liwei Chan. 2018. FacePush. In *The 31st Annual ACM Symposium on User Interface Software and Technology - UIST '18*. ACM Press. <https://doi.org/10.1145/3242587.3242588>
- [17] Xiang 'Anthony' Chen, Ye Tao, Guanyun Wang, Runchang Kang, Tovi Grossman, Stelian Coros, and Scott E. Hudson. 2018. Forte. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. <https://doi.org/10.1145/3173574.3174070>
- [18] Pei-Yu Chi. 2016. *Designing Video-Based Interactive Instructions*. Ph.D. Dissertation. University of California, Berkeley.
- [19] Pei-Yu (Peggy) Chi, Daniel Vogel, Mira Dontcheva, Wilmot Li, and Björn Hartmann. 2016. Authoring Illustrations of Human Movements by Iterative Physical Demonstration. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (Tokyo, Japan) (*UIST '16*). ACM, New York, NY, USA, 809–820. <https://doi.org/10.1145/2984511.2984559>
- [20] Pei-Yu (Peggy) Chi, Daniel Vogel, Mira Dontcheva, Wilmot Li, and Björn Hartmann. 2016. Authoring Illustrations of Human Movements by Iterative Physical Demonstration. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (Tokyo, Japan) (*UIST '16*). Association for Computing Machinery, New York, NY, USA, 809–820. <https://doi.org/10.1145/2984511.2984559>
- [21] Christian Corsten, Simon Voelker, Andreas Link, and Jan Borchers. 2018. Use the Force Picker, Luke. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. <https://doi.org/10.1145/3173574.3174235>
- [22] James E Cutting. 2002. Representing Motion in a Static Image: Constraints and Parallels in Art, Science, and Popular Culture. *Perception* 31, 10 (2002), 1165–1193. <https://doi.org/10.1068/p3318> PMID: 12430945.
- [23] Alix Ducros, Clemens N. Klokose, and Aurélien Tabard. 2019. Situated Sketching and Enactment for Pervasive Displays. In *Proceedings of the 2019 ACM International Conference on Interactive Surfaces and Spaces* (Daejeon, Republic of Korea) (*ISS '19*). Association for Computing Machinery, New York, NY, USA, 217–228. <https://doi.org/10.1145/3343055.3359702>
- [24] J. C. Dunn. 1973. A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. *Journal of Cybernetics* 3, 3 (1973), 32–57. <https://doi.org/10.1080/01969727308546046> arXiv:<https://doi.org/10.1080/01969727308546046>
- [25] Elsevier. 2020. Graphical abstracts. Retrieved June 9th, 2020 from <https://www.elsevier.com/authors/journal-authors/graphical-abstract>
- [26] Joseph L. Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin* 76, 5 (1971), 378–382. <https://doi.org/10.1037/h0031619> Place: US Publisher: American Psychological Association.
- [27] Malcolm Fleming. 1967. Classification and Analysis of Instructional Illustrations. *AV Communication Review* 15, 3 (1967), 246–258. <http://www.jstor.org/stable/30217383>
- [28] Euan Freeman, Julie Williamson, Sriram Subramanian, and Stephen Brewster. 2018. Point-and-Shake. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. <https://doi.org/10.1145/3173574.3173592>
- [29] Emmanouil Giannisakis, Gilles Bailly, Sylvain Malacria, and Fanny Chevalier. 2017. IconHK: Using Toolbar Button Icons to Communicate Keyboard Shortcuts. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) (*CHI '17*). Association for Computing Machinery, New York, NY, USA, 4715–4726. <https://doi.org/10.1145/3025453.3025595>
- [30] Alix Goguy, Sylvain Malacria, and Carl Gutwin. 2018. Improving Discoverability and Expert Performance in Force-Sensitve Text Selection for Touch Devices with Mode Gauges. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. <https://doi.org/10.1145/3173574.3174051>
- [31] Saul Greenberg, Sheelagh Carpendale, Nicolai Marquardt, and Bill Buxton. 2011. *Sketching User Experiences: The Workbook* (1st ed.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- [32] Tovi Grossman, Fanny Chevalier, and Rubaiat Habib Kazi. 2015. Your Paper is Dead! Bringing Life to Research Articles with Animated Figures. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems* (Seoul, Republic of Korea) (*CHI EA '15*). Association for Computing Machinery, New York, NY, USA, 461–475. <https://doi.org/10.1145/2702613.2732501>
- [33] Jan Gugenheimer, Evgeny Stemasov, Harpreet Sareen, and Enrico Rukzio. 2018. FaceDisplay. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. <https://doi.org/10.1145/3173574.3173628>
- [34] Y. Hansen. 2000. *Visualization for Thinking, Planning, and Problem Solving*. 193–220 pages. <https://mediosavanzados.files.wordpress.com/2007/06/visualization-for-thinking.pdf>
- [35] S. Haroz, R. Kosara, and S. L. Franconeri. 2016. The Connected Scatterplot for Presenting Paired Time Series. *IEEE Transactions on Visualization and Computer Graphics* 22, 9 (2016), 2174–2186.
- [36] Kasper Hornbæk and Antti Oulasvirta. 2017. What Is Interaction?. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) (*CHI '17*). ACM, New York, NY, USA, 5040–5052. <https://doi.org/10.1145/3025453.3025765>
- [37] Jin Huang, Feng Tian, Xiangmin Fan, Xiaolong (Luke) Zhang, and Shumin Zhai. 2018. Understanding the Uncertainty in 1D Unidirectional Moving Target Selection. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) (*CHI '18*). Association for Computing Machinery, New York, NY, USA, 1–12. <https://doi.org/10.1145/3173574.3173811>
- [38] Jessica Hullman and Benjamin Bach. 2018. Picturing Science: Design Patterns in Graphical Abstracts. In *International Conference on Theory and Application of Diagrams*. Springer.
- [39] Ke Huo, Yuanzhi Cao, Sang Ho Yoon, Zhuangying Xu, Guiming Chen, and Karthik Ramani. 2018. Scenariot. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. <https://doi.org/10.1145/3173574.3173811>

- 3173574.3173793
- [40] Ke Huo, Tianyi Wang, Luis Paredes, Ana M. Villanueva, Yuanzhi Cao, and Karthik Ramani. 2018. SynchronizAR. In *The 31st Annual ACM Symposium on User Interface Software and Technology - UIST '18*. ACM Press. <https://doi.org/10.1145/3242587.3242595>
- [41] Yunwoo Jeong, Han-Jong Kim, and Tek-Jin Nam. 2018. Mechanism Perfboard. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. <https://doi.org/10.1145/3173574.3173985>
- [42] Robert E. Kahn and John H. Cafarella. 1998. Unconstrained pointing interface for natural human interaction with a display-based computer system. <https://patentimages.storage.googleapis.com/e0/a6/26/bb6514edf9149e/US5793361.pdf> US Patent 5,793,361.
- [43] Rubaiat Habib Kazi, Fanny Chevalier, Tovi Grossman, and George Fitzmaurice. 2014. Kitty: Sketching Dynamic and Interactive Illustrations. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (Honolulu, Hawaii, USA) (UIST '14). Association for Computing Machinery, New York, NY, USA, 395–405. <https://doi.org/10.1145/2642918.2647375>
- [44] Rubaiat Habib Kazi, Fanny Chevalier, Tovi Grossman, Shengdong Zhao, and George Fitzmaurice. 2014. Draco: Bringing Life to Illustrations with Kinetic Textures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) (CHI '14). Association for Computing Machinery, New York, NY, USA, 351–360. <https://doi.org/10.1145/2556288.2556987>
- [45] Mohamed Khamis, Ludwig Trotter, Ville M kela, Emanuel von Zeszschwitz, Jens Le, Andreas Bulling, and Florian Alt. 2018. CueAuth. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 4 (dec 2018), 1–22. <https://doi.org/10.1145/3287052>
- [46] Rushil Khurana, Karan Ahuja, Zac Yu, Jennifer Mankoff, Chris Harrison, and Mayank Goel. 2018. GymCam. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 4 (dec 2018), 1–17. <https://doi.org/10.1145/3287063>
- [47] Thomas Kosch, Mariam Hassib, Paweł W. Woźniak, Daniel Buschek, and Florian Alt. 2018. Your Eyes Tell. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. <https://doi.org/10.1145/3173574.3174010>
- [48] Klaus Krippendorff. 1970. Estimating the Reliability, Systematic Error and Random Error of Interval Data. *Educational and Psychological Measurement* 30, 1 (1970), 61–70. <https://doi.org/10.1177/001316447003000105> arXiv:<https://doi.org/10.1177/001316447003000105>
- [49] J. Richard Landis and Gary G. Koch. 1977. The Measurement of Observer Agreement for Categorical Data. *Biometrics* 33, 1 (1977), 159–174. <http://www.jstor.org/stable/2529310>
- [50] Jonathan Lazar, Jinjuan Heidi Feng, and Harry Hochheiser. 2010. *Research Methods in Human-Computer Interaction*. Wiley Publishing.
- [51] Huy Viet Le, Thomas Kosch, Patrick Bader, Sven Mayer, and Niels Henze. 2018. PalmTouch. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. <https://doi.org/10.1145/3173574.3173934>
- [52] Joon Hyub Lee, Sang-Gyun An, Yongkwan Kim, and Seok-Hyung Bae. 2018. Projective Windows. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. <https://doi.org/10.1145/3173574.3173792>
- [53] Rong-Hao Liang, Meng-Ju Hsieh, Jheng-You Ke, Jr-Ling Guo, and Bing-Yu Chen. 2018. RFIMatch. In *The 31st Annual ACM Symposium on User Interface Software and Technology - UIST '18*. ACM Press. <https://doi.org/10.1145/3242587.3242620>
- [54] Hyunchul Lim, Jungmin Chung, Changhoon Oh, SoHyun Park, Joonhwan Lee, and Bongwon Suh. 2018. Touch+Finger. In *The 31st Annual ACM Symposium on User Interface Software and Technology - UIST '18*. ACM Press. <https://doi.org/10.1145/3242587.3242651>
- [55] Wanyu Liu, Olivier Rioul, Joanna McGrenere, Wendy E. Mackay, and Michel Beaudouin-Lafon. 2018. BIGFile. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. <https://doi.org/10.1145/3173574.3173959>
- [56] Pedro Lopes, Sijing You, Alexandra Ion, and Patrick Baudisch. 2018. Adding Force Feedback to Mixed Reality Experiences and Games using Electrical Muscle Stimulation. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. <https://doi.org/10.1145/3173574.3174020>
- [57] Nicolai Marquardt, Frederik Brudy, Can Liu, Ben Bengler, and Christian Holz. 2018. SurfaceConstellations. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. <https://doi.org/10.1145/3173574.3173928>
- [58] Nicolai Marquardt, Ken Hinckley, and Saul Greenberg. 2012. Cross-Device Interaction via Micro-Mobility and f-Formations. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology* (Cambridge, Massachusetts, USA) (UIST '12). Association for Computing Machinery, New York, NY, USA, 13–22. <https://doi.org/10.1145/2380116.2380121>
- [59] Patricia I. Cornelio Martinez, Emanuela Maggioni, Kasper Hornb k, Marianna Obrist, and Sriram Subramanian. 2018. Beyond the Libet Clock. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. <https://doi.org/10.1145/3173574.3174115>
- [60] Sebastian Marwecki and Patrick Baudisch. 2018. Scenograph. In *The 31st Annual ACM Symposium on User Interface Software and Technology - UIST '18*. ACM Press. <https://doi.org/10.1145/3242587.3242648>
- [61] Sebastian Marwecki, Maximilian Brehm, Lukas Wagner, Lung-Pan Cheng, Florian 'Floyd' Mueller, and Patrick Baudisch. 2018. VirtualSpace - Overloading Physical Space with Multiple Virtual Reality Users. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. <https://doi.org/10.1145/3173574.3173815>
- [62] Damien Masson, Sylvain Malacria, Edward Lank, and G ry Casiez. 2020. Chameleon: Bringing Interactivity to Static Digital Documents. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) (CHI '20). Association for Computing Machinery, New York, NY, USA, 1–13. <https://doi.org/10.1145/3313831.3376559>
- [63] Sven Mayer, Valentin Schwind, Robin Schweigert, and Niels Henze. 2018. The Effect of Offset Correction and Cursor on Mid-Air Pointing in Real and Virtual Environments. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. <https://doi.org/10.1145/3173574.3174227>
- [64] Erin McAweeney, Haihua Zhang, and Michael Nebeling. 2018. User-Driven Design Principles for Gesture Representations. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. <https://doi.org/10.1145/3173574.3174121>
- [65] Scott McCloud. 1993. *Understanding Comics: The Invisible Art*. Tundra Publishing Ltd.
- [66] Nora McDonald, Sarita Schoenebeck, and Andrea Forte. 2019. Reliability and Inter-rater Reliability in Qualitative Research: Norms and Guidelines for CSCW and HCI Practice. *Proc. ACM Hum.-Comput. Interact.* 3, CSCW, Article 72 (Nov. 2019), 23 pages. <https://doi.org/10.1145/3359174>
- [67] Fabrizio Pece, Juan Jose Zarate, Velko Vechev, Nadine Besse, Olexandr Gudozhnik, Herbert Shea, and Otmar Hilliges. 2017. MagTics: Flexible and Thin Form Factor Magnetic Actuators for Dynamic and Wearable Haptic Feedback. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology* (Qu bec City, QC, Canada) (UIST '17). Association for Computing Machinery, New York, NY, USA, 143–154. <https://doi.org/10.1145/3126594.3126609>
- [68] Eujin Pei, R.I. Campbell, and Mark Evans. 2011. A Taxonomic Classification of Visual Design Representations Used by Industrial Designers and Engineering Designers. *The Design Journal* 14 (2011), 64–91. <https://doi.org/10.2752/175630610X12877385838803>
- [69] David Petrou. 2012. Head-mounted display that displays a visual representation of physical interaction with an input interface located outside of the field of view. <https://patentimages.storage.googleapis.com/5a/23/22/d3db8fd8048482/US8217856.pdf> US Patent 8,217,856 B1.
- [70] Henning Pohl and Kasper Hornb k. 2018. ElectricTch. In *The 31st Annual ACM Symposium on User Interface Software and Technology - UIST '18*. ACM Press. <https://doi.org/10.1145/3242587.3242647>
- [71] Ivan Poupyrev and Gaetano Roberto Aiello. 2015. Radar-based gesture-recognition through a wearable device. <https://patentimages.storage.googleapis.com/77/69/61/d96948a8b469ba/US20150346820A1.pdf> US Patent 2015/0346820A1.
- [72] Ivan Poupyrev and Shigeaki Maruyama. 2011. Apparatus and method for touch screen interaction based on tactile feedback and pressure measurement. <https://patentimages.storage.googleapis.com/b4/7e/9f/37227d90b5fa6c/US7952566B2.pdf> US Patent 7,952,566 B2.
- [73] Abbas Rafii. 2014. Two-dimensional method and system enabling three-dimensional user interaction with a device. <https://patentimages.storage.googleapis.com/4b/15/08/06d12780bfec5d/US8686943.pdf> US Patent 8,686,943 B1.
- [74] Nimesha Ranasinghe, Adrian David Cheok, Owen Noel Newton Fernando, Hideaki Nii, and Gopalakrishnakone Ponnampalam. 2011. Electronic Taste Stimulation. In *Proceedings of the 13th International Conference on Ubiquitous Computing* (Beijing, China) (UbiComp '11). Association for Computing Machinery, New York, NY, USA, 561–562. <https://doi.org/10.1145/2030112.2030213>
- [75] Nimesha Ranasinghe, Adrian David Cheok, and Ryohei Nakatsu. 2012. Taste/IP: The Sensation of Taste for Digital Communication. In *Proceedings of the 14th ACM International Conference on Multimodal Interaction* (Santa Monica, California, USA) (ICMI '12). Association for Computing Machinery, New York, NY, USA, 409–416. <https://doi.org/10.1145/2388676.2388768>
- [76] Oliver Schneider, Jotaro Shigeyama, Robert Kovacs, Thijs Jan Roumen, Sebastian Marwecki, Nico Boeckhoff, Daniel Amadeus Gloeckner, Jonas Bounama, and Patrick Baudisch. 2018. DualPanto. In *The 31st Annual ACM Symposium on User Interface Software and Technology - UIST '18*. ACM Press. <https://doi.org/10.1145/3242587.3242604>
- [77] Youngbo Aram Shim, Jaeyeon Lee, and Geehyuk Lee. 2018. Exploring Multimodal Watch-back Tactile Display using Wind and Vibration. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. <https://doi.org/10.1145/3173574.3173706>
- [78] Nikos Sidiropoulos, Sina Hadi Sohi, Thomas Lin Pedersen, Bo Torben Porse, Ole Winther, Nicolas Rapin, and Frederik Otzen Bagger. 2018. SinaPlot: an enhanced chart for simple and truthful representation of single observations over multiple

- classes. *Journal of Computational and Graphical Statistics* 27, 3 (2018), 673–676.
- [79] Alexa F. Siu, Eric J. Gonzalez, Shenli Yuan, Jason B. Ginsberg, and Sean Follmer. 2018. shapeShift. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. <https://doi.org/10.1145/3173574.3173865>
- [80] Kinga Skorupska, Manuel Nunez, Wieslaw Kopec, and Radoslaw Nielek. 2018. Older Adults and Crowdsourcing: Android TV App for Evaluating TEDx Subtitle Quality. *Proc. ACM Hum.-Comput. Interact.* 2, CSCW, Article 159 (Nov. 2018), 23 pages. <https://doi.org/10.1145/3274428>
- [81] Brian A. Smith and Shree K. Nayar. 2018. The RAD. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. <https://doi.org/10.1145/3173574.3174090>
- [82] Ryo Suzuki, Jun Kato, Mark D. Gross, and Tom Yeh. 2018. Reactile. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. <https://doi.org/10.1145/3173574.3173773>
- [83] Ryo Suzuki, Junichi Yamaoka, Daniel Leithinger, Tom Yeh, Mark D. Gross, Yoshihiro Kawahara, and Yasuaki Kakehi. 2018. Dynablock. In *The 31st Annual ACM Symposium on User Interface Software and Technology - UIST '18*. ACM Press. <https://doi.org/10.1145/3242587.3242659>
- [84] Shan-Yuan Teng, Tzu-Sheng Kuo, Chi Wang, Chi huan Chiang, Da-Yuan Huang, Liwei Chan, and Bing-Yu Chen. 2018. PuPoP. In *The 31st Annual ACM Symposium on User Interface Software and Technology - UIST '18*. ACM Press. <https://doi.org/10.1145/3242587.3242628>
- [85] M. Tovey, S. Porter, and R. Newman. 2003. Sketching, concept development and automotive design. *Design Studies* 24, 2 (2003), 135 – 153. [https://doi.org/10.1016/S0142-694X\(02\)00035-2](https://doi.org/10.1016/S0142-694X(02)00035-2)
- [86] Daniel Trindade, André Rodrigues, Tiago Guerreiro, and Hugo Nicolau. 2018. Hybrid-Braille. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. <https://doi.org/10.1145/3173574.3173601>
- [87] Erick Tseng, Daniel Johansson, Michael A. Cleron, and Indika C. Mendis. 2015. Long press interface interactions. <https://patentimages.storage.googleapis.com/1a/34/db/59a5f539644bce/US8954887.pdf> US Patent 8,954,887 B1.
- [88] USPTO. 2018. Drawings section. Retrieved May 25th, 2020 from <https://www.uspto.gov/web/offices/pac/mpep/s1825.html>
- [89] Claus O Wilke. 2019. *Fundamentals of data visualization: a primer on making informative and compelling figures*. O'Reilly Media.
- [90] Anusha Withana, Daniel Groeger, and Jürgen Steimle. 2018. Tacttoo. In *The 31st Annual ACM Symposium on User Interface Software and Technology - UIST '18*. ACM Press. <https://doi.org/10.1145/3242587.3242645>
- [91] Pui Chung Wong, Kening Zhu, and Hongbo Fu. 2018. FingerT9. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. <https://doi.org/10.1145/3173574.3173752>
- [92] Yu-Chian Wu, Te-Yen Wu, Paul Taelle, Bryan Wang, Jun-You Liu, Pin sung Ku, Po-En Lai, and Mike Y. Chen. 2018. ActiveErgo. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. <https://doi.org/10.1145/3173574.3174132>
- [93] Haijun Xia, Nathalie Henry Riche, Fanny Chevalier, Bruno De Araujo, and Daniel Wigdor. 2018. DataInk. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press. <https://doi.org/10.1145/3173574.3173797>
- [94] Keng-Ta Yang, Chiu-Hsuan Wang, and Liwei Chan. 2018. ShareSpace. In *The 31st Annual ACM Symposium on User Interface Software and Technology - UIST '18*. ACM Press. <https://doi.org/10.1145/3242587.3242630>
- [95] Jingjie Zheng, Blaine Lewis, Jeff Avery, and Daniel Vogel. 2018. FingerArc and FingerChord. In *The 31st Annual ACM Symposium on User Interface Software and Technology - UIST '18*. ACM Press. <https://doi.org/10.1145/3242587.3242589>

A TAXONOMY OVERALL STATISTICS

We analysed the global tendencies of codes within the dataset. We chose to exclude patent figures from the analysis since they seem to always use the same codes³. In the end, we analysed the codes applied to exactly 594 figures coming from the proceedings part of our dataset.

A.1 Appearance of figures

387 figures (65%) used photographs while 175 figures (28%) used drawings. Note that both types are not exclusive and the use of both

³This is unsurprising given that illustrations for patents must follow the rules mentioned section 2.2.5. Yet, we will discuss the strategies used for patents in section 4

photographs and drawings was found in 28 (5%) figures. We only

found 71 figures (11%) using a drawn UI, and 142 (23%) using a *screen capture*, which represents a total of 213 (36%) figures including a UI. Among these 213 figures including UIs, the UI was the main component of the figure for 117 (20%) of them. 555 figures (93%) used *colours*, 26 (4%) were *monochrome* and 20 (3%) used a *greyscale*.

A.2 Purpose and structure.

We found respectively 250 (42%), 232 (39%) and 171 (28%) figures representing *design spaces*, *interactive systems* and *interaction sequences*. The structure used seems to be correlated to the represented content.

Design space. A design space is a figure illustrating a set of functionalities, possibilities, commands or use cases available in a system. 229 figures (91% of 250) were composed of multiple independent frames, *i.e.* there is no relationship between the frames. Within these 229 figures, 193 (84% of 229) used a *fixed time* through the frames (we refer to the supplementary materials for a definition of the codes used).

Interactive system. A figure showing an interactive system focuses on the setup of a system and its sensing capabilities rather than on the interaction between the user and the system. 136 figures (58% of 232) focused on a system, using most of the time a single frame. Within these 136 figures, 129 (94% of 136) used a *fixed time* through the frames as well.

Interaction sequence. An interaction sequence is a sequence of actions that must be accomplished in a specific order to launch a system command. 145 figures (84% of 171) were composed of multiple dependent frames, *i.e.* there is a relationship between the frames. For these 145 figures, time was typically progressing through the different frames in 127 of them (87% of 145).

A.3 Dynamic.

We found 170 figures (28%) representing *time evolving*. Interestingly, within these 170 figures, 147 (86%) illustrate an interaction sequence. Evolution of time is illustrated using three main techniques.

Multiple frames. 141 (82% of 170) used *multiple frames* to illustrate the progression of time across the figure. To refer to each frame, 97 figures (68% of 141) used letters, 23 figures (16% of 141) titles and 9 (6% of 141) numbers.

Movement effects. 41 (24% of 170) used a *stroboscopic effect* to represent a *time progression*. This little number can be explained as it requires better drawing and figure design skills to produce a *stroboscopic effect*.

Lines and arrows. We also found an important use of lines and arrows to represent movements (52 figures, 30% of 170), trajectories (41 figures, 24% of 170) and transitions (24 figures, 14% of 170).

A.4 User representation.

We found a total of 558 figures (96%) illustrating user(s) which leaves 23 figures where no user is represented (*e.g.* interactive systems only).

Number of users: 491 figures (88% of 558) represented only one user and only 67 figures (12% of 558) multiple users.

Body part. Unsurprisingly, the most frequently represented *body part* is the *hand* with 302 figures (50%). This can be easily explained as the most represented types of interaction are *touch interaction*

(192 figures, 32%) and *tangible interaction* (152 figures, 25%). Then, we did find representations of *users' upper body*, *full body* and *fingers* with corresponding proportions of 130 (21%), 85 (14%) and 73 figures (12%).

Anonymisation. Surprisingly, we only found 9 figures (1%) over the whole dataset using user anonymisation technique (using *blur effect* in our case).

Point of view. The most frequently used point of view was *3rd person* with 323 figures (54%). Within the 323 figures, 254 (78% of 323) illustrate a single user and 52 figures (16% of 323) multiple users. The second most used point of view was *1st person* view with 170 figures (29%), mostly used to illustrate one user only (152 figures, 89% of 170).