**Proceedings of the ASME 2024**
**International Design Engineering Technical Conferences and Computers and Information in Engineering Conference**
**IDETC/CIE2024**
**August 25-28, 2024, Washington, DC**

# DETC2024-143806

# UNDERSTANDING COMPLEX SKETCH RECOGNITION STRATEGIES FOR INTELLIGENT SKETCH-BASED DESIGN TOOLS

**Gaëlle Baudoux**
University of California, Berkeley
Berkeley, CA
Email: gbaudoux@berkeley.edu

**Kosa Goucher-Lambert**
University of California, Berkeley
Berkeley, CA
Email: kosa@berkeley.edu

**ABSTRACT**

*Despite recent advances in multi-modal AI tools (e.g., tools leveraging text-to-image models), there is a significant gap in the ability of such systems to be incorporated into complex design and engineering work. This gap is further exacerbated in contexts where sketch-based inputs are desirable due to the difficulty in recognizing freehand sketches or interpreting underlying human intent. To better surface requirements for emerging sketch-based AI systems for complex design context, we consider a case study involving architectural design; this is a domain for which, to our knowledge, there have been no architectural sketch-based AI tools that recognize freely produced plans or perspectives for downstream applications, including generating inspirational images. Using a Wizard of Oz experimental paradigm, we substitute the "tool" with human agents and conduct a lab-based study in which professional architects complete a design brief using this "tool". Results demonstrate that human agents not only rely on visible sketch elements (i.e., lines) and architectural drawing codes, but also on their memory of previous lines and their knowledge of the design brief to comprehend perceived lines. In addition to gradually developing an understanding of the designed artifact, human agents also construct an understanding of the designer's intentions. These activities are crucial for the agent to obtain a functional model of the designed object, beyond a purely topological and geometric perception model. Insights about this human workflow bring new potential techniques of sketch recognition for design tasks, informing the inclusion of new resources and software within AI tools.*

Keywords: Design aid, Analogical reasoning, Sketch-based AI, Sketch recognition, Wizard of Oz.

## 1. INTRODUCTION

Preliminary design phases define a significant portion of the final performance of a designed artifact, impacting both the economy and ecology of a project [1]. Therefore, it is crucial to make appropriate design choices from the early phases. To meet this need, we work on design aid to improve preliminary design, specifically in generating and evaluating solutions. A promising approach, according to several researchers [2-8], is to use creative ideation and automate analogical reasoning. Indeed, idea generation can be driven by analogical reasoning, a recognized powerful design strategy that has been studied extensively over the last 20 years, with reemerging interest in recent work [3-7]. Visual analogies, in particular, can improve design quality and the performance of proposed solutions [8], as well as enhancing creativity by overcoming the fixation problem. This cognitive strategy (Fig. 1) involves pairing an inspirational source and a characteristic of the artifact to be designed, and then transferring certain properties of the source-object to integrate them into the designed object [2].

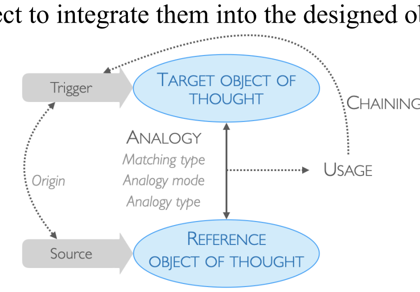


**FIGURE 1:** Pairing scheme in analogy [2].

In this pathway, AI image generators are a promising tool to support creative activity. These generative software programs (e.g. Midjourney or DALL-E) produce images based on text commands, known as prompts [9]. Although text-to-image generators offer potential for augmenting ideation, they have limitations. These generators only work on the basis of text prompts, and pausing to write a command to receive images may disrupt the designer's flow of thought when sketching. Furthermore, it has been observed by researchers that formulating accurate textual prompts can be challenging in practice. This limitation affects the suitability of the received images to the designers' requirements [9,10]. Therefore, we propose to investigate the potential of sketch-based generative AI tools that work with sketched inputs, instead of textual prompts, as complex as whole plans or elevation perspectives to generate the inspirational images for designers.

Despite recent advances in ideation stimulation tools and generative AI, there is still a significant need in the complex design domain where freehand sketches remain the primary medium of ideation. The architectural sector, in particular, lacks tools to assist in solution generation and evaluation during ideation phases [11]. This domain presents an interesting case study for improving multi-modal AI tools and bridging the gap in the need for sketch-based tools with complex and freely produced sketched inputs. We thus envision a future tool for complex design environment (e.g. architecture) that suggests inspirational images, based on the actual drawings used in these creative phases, to improve generation and evaluation of ideas.

In our previous work [12,13], we demonstrated the added value of using sketch input for non-disruptive tools. The design activity was indeed studied under sketch-based inspirational stimuli, and it was demonstrated that sketch-based AI tools retain the well-known benefits of generative AI for ideation while overcoming their limitations by sending images that are more accurate to the object designed and with no disruption of the design flow. Sketch-based generated images were also used for larger activities than only idea generation. However, we now need to study how to achieve the recognition of design sketches. As such, this study is positioned prior to any prototype development, and the focus in this paper is on the human agent's sketch recognition activity. The purpose of this paper is to explore the challenges of building a tool for architectural design sketch recognition and inspirational image generation, with the goal of gaining a better understanding of the necessary inputs, including data and rules, to improve architectural line-by-line sketch recognition. To accomplish this, we assume that observing the human workflow in performing the task of interpreting architectural line-by-line sketches could provide the necessary information. Therefore, we aim to answer the two following research questions:

- *What resources do human agents use to understand the architectural object under design?*
- *What is their work procedure from receiving graphic features from line-by-line sketches to constructing a mental model of the design?*

To gain insight into the functionality of sketch-based generative AI tools, this paper begins with some key elements of current research on sketch-based AI tools for architectural design. This establishes what is currently possible and what does not yet exist (section 2). Next, a Wizard of Oz experiment is set up, substituting the tool with human agents, to observe the human workflow in performing the architectural sketch recognition task (section 3). By understanding the resources and cognitive strategies used by human agents to interpret complex sketches, we provide insights about procedures and rules, but perhaps more interestingly about the knowledge bases mobilized, as well as the challenges involved with such a complex task of sketch understanding (see section 4). The research outcomes will be valuable for various design domains, offering new possibilities for shaping AI tool workflows when textual inputs are not applicable, and other modalities of interaction would be more applicable (e.g., sketches), while keeping human agency over the interactive process intact.

## 2. CURRENT SKETCH-BASED AI TOOLS

This section investigates each of the proposed tools' three features - sketch as an input (section 2.1), recognition of more specifically architectural sketches (section 2.2), and inspirational images as an output (section 2.3) - before synthesizing the general issue (section 2.4).

### 2.1 Sketch-based tools in design domains

Sketch-based tools have been studied for a long time in academic design research and have evolved to understand more and more hand drawings for a wide range of applications.

In the late 1990s, mechanical engineering researchers began developing design aids based on sketches, such as SketchIt [14], ASSIST [15], and UDSI [16]. These technologies were capable of interpreting line pixels to generate geometric shapes and abstract drawings by combining direction and speed information [14, 15]. They could also recognize text, geometric shapes, arrows, and expected symbols [14-16]. Thus, they were able to comprehend a drawing, and descriptions of the desired behavior, in the case of ASSIST or SketchIt, to generate the corresponding component. In mechanical engineering, it was even possible to suggest variations of these components [14].

Over the years, different types of input have been developed, such as 2D drawings in plane, 2D drawings in multiple specified planes forming a 3D space, immersion drawings in a 3D model, or perspective drawings [17]. However, the initial SketchIt, ASSIST or UDSI tools were only capable of recognizing simple, clean-lined drawings composed of basic geometric shapes and pre-encoded symbols [14-16].

More recently, to support visual and multi-modal design-by-analogy in the engineering design process, Jiang et al. encourage the development of novel tools to process non-textual inputs such as sketches, images, or 3D models [6]. Zhang and Jin propose a framework for the search and retrieval of visual stimuli to enable the discovery of visual analogies from large datasets of design materials (e.g., sketches, CAD drawings, photographs, etc.) based on designers' initial sketches [18]. They demonstrate how visually related sketches to a designer's sketch-based input can be discovered to support visual analogy [19]. Kim et al. also develop a co-creative sketching AI partner to provide inspirational sketches based on visual and conceptual similarity to a designer's sketch [20]. Arora et al. [21] developed a sketch-based tool that generates new sets of inspirational sketches based on input images of rough sketches from the designer. Some tools can additionally recognize motion significance arrows and propose 3D-model solutions that meet sketched mechanical constraints [22]. Image-based search has also been explored by Jiang et al. to retrieve visually relevant patent images [23] and by Kwon et al. to discover alternative uses for products [24]. Beyond tools that support image and sketch inputs, Kwon et al. built a multi-modal platform to retrieve 3D-model parts based on similarities in visual and functional features to 3D-modeled inputs specified by the designer [25].

However, these tools are not designed to manage the complexity, amount of information, and vagueness of architectural sketches. The sketches used in these tools are clean and unambiguous (see Fig. 2).

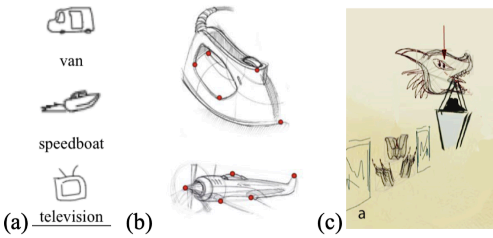


**FIGURE 2:** Type of sketches managed by recent tools [19,21,22].

## 2.2 Architectural sketch recognition feasibility

In the field of architectural design, some researchers have attempted to address the task of recognizing architectural sketches, regardless of what these tools are used for. The tools developed for recognizing architectural sketches differ slightly from other design tools mentioned previously. In addition to the sketch recognition strategies used in engineering design, these tools include disambiguation steps and are trained to recognize typical architectural drawing codes. They also typically recognize written characters to understand room labels and common annotations.

These tools include Valveny and Marti's tool [26], Lee et al. 's tool [27], Sketch It Make It [28], SolidSketch [29], EsQUIsE [30], and NEMo [31]. EsQUIsE, developed by Leclercq, Juchmes, and Azar [30], utilizes a multi-agent system to interpret the sketched plans and generate a 3D model of the building (refer to Fig. 3) as well as a semantic model of the plan, then used to perform simulations and evaluations of constructive applications [30]. Addressing the limitations of EsQUIsE, NEMo [31] recognizes freehand drawings using conventional symbols commonly found in architecture and generates a 3D model of the building.

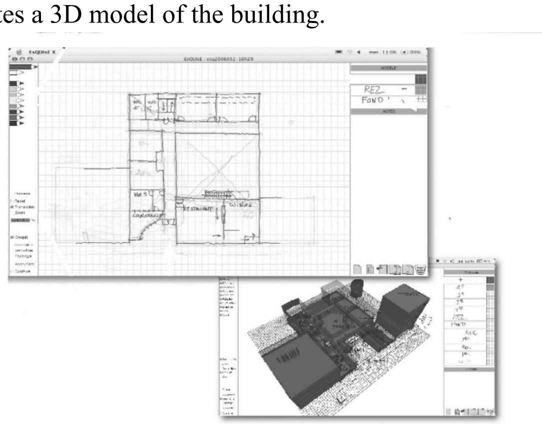

**FIGURE 3:** Illustration of EsQUIsE [30].

These prototypes demonstrate that architectural sketch recognition is currently achieved by limiting the drawing process to conform to drawing codes that can be understood by the software. The most advanced sketch recognition prototype to date, to our knowledge, is NEMo, which was developed in the 2010s. No more recent or advanced documented similar prototypes for architecture were found. Recent research on sketch recognition has mainly concentrated rather on software architecture and the performance of neural or deep learning systems. It is only applied to tasks that involve interpreting drawings of everyday objects, such as SketchPointNet by Wang [32] or SketchGAN by Liu [33]. These methods are not currently utilized in design. Moreover, the recognition is typically based on CAD plans or 3D models rather than freehand sketches.

Regarding the output, the current architectural softwares can produce 2D and/or 3D models based on the interpretation of sketches, resulting in 2D representations, 3D models, and semantic diagrams. Assembly techniques involve modeling geometric shapes and positioning them in relation to each other, modeling scenes composed of semantically recognized components, and modeling 3D meshes [17]. However, none of these techniques produce inspiring images as output.

## 2.3 Generative AI tools for image generation

When searching for sketch recognition tools that generate images, some tools are designed to aid in the ideation process by providing either inspirational or rendered images.

Instrumented co-creation was already being studied around or even before 2015, prior to the recent surge in AI image generators. For example, the Electronic Cocktail Napkin [34] retrieves and displays architectural components related to the designer's sketch. Drawing Apprentice [35] is a sketching support tool that responds to the designer's sketch by sending a similar sketch, thus maintaining engagement in design. Sentient SketchBook [36] and 3Buddy [37] are two design tools aiming to improve the designer's exploration of the solution-space through ideation human-machine conversations. They provide more goal-oriented accurate outputs. Two recent sketch-based tools for co-creation by image generation are of interest:

Sketch2Pix (Fig. 4) is an interactive application that supports architectural sketching augmented by an automated image-to-image translation process [38]. Designers can sketch using augmented brushes that translate strokes into pre-programmed images. For instance, they can quickly create a perspective sketch by using pre-trained brushes like 'fence' or 'hedge' to draw rendered fences or hedges. The Creative Sketching Partner (Fig. 5 [39]) and the similar Collaborative Ideation Partner [20] are interactive systems that recognize a current design sketch and propose a response sketch (CSP) or an image (CIP) from another category or domain that shares some structural or semantic aspects. The response sketch is modulated by specified level of visual/conceptual similarity.

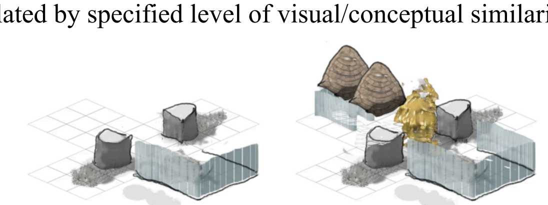

**FIGURE 4:** Composition of a sketch using preset brushes [38].

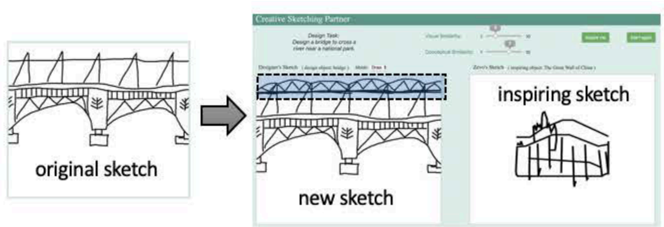


**FIGURE 5:** Examples of participants' new sketch based on inspiring sketch [39, p. 225].

### 2.4 General issue

This background synthesis revealed a lack of tools that utilize sketch-based Generative AI capable of understanding typically complex and non-pre-coded architectural drawings (e.g. Fig. 6) while also providing designers with inspirational images. Furthermore, most AI image generators currently utilize LLMs and require text as input. Large Language Models (LLMs) have shown effectiveness in processing images, but they are not yet capable of processing 'line-by-line' sketches. Current results are still in the early stages [40]. The current tools are performative, but they lack alignment with the characteristics and specificities of architectural sketches. This raises the general question of what makes the task of recognizing architectural design sketches so difficult.

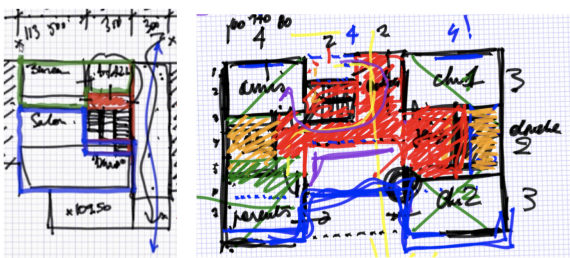

**FIGURE 6:** Examples of typical architectural ideation sketches.

## 3. MATERIALS AND METHODS

This section describes the experiment conducted to simulate the task of architectural sketch recognition with human agents. It begins by explaining the global experiment and the task, as well as the implementation of the physical space, data collection, and coding.

### 3.1 General experiment procedure

To better understand the challenges of this proposal, and to develop an ecologically valid aid, we seek to recreate a realistic task context. Instead of providing pre-selected sketches to human agents, we simulate a design situation where professional architects create a single-family house project according to a given design brief and sketch it on a tablet with drawing software designed to resemble natural drawing, as they would in a normal design scenario, and interact with the "intelligent tool" for 1h30. The tool manages dynamic, evolving, realistic sketches. Designers are provided with live inspirational images generated based on their sketches.

We therefore mobilize the principle of the Wizard of Oz to simulate this sketch-based tool. The Wizard of Oz is a technique that consists of simulating the functionalities of an innovative technology by replacing them with equivalent human work, hidden and in real time. In this way, the observed subject believes that he/she is using the so-called technology without the need for it to be developed. This makes it possible to assess in advance its impact on users and their interaction with the machine [41] and thus help to figure out the development needs. Figure 7 shows the Wizard of Oz set up.

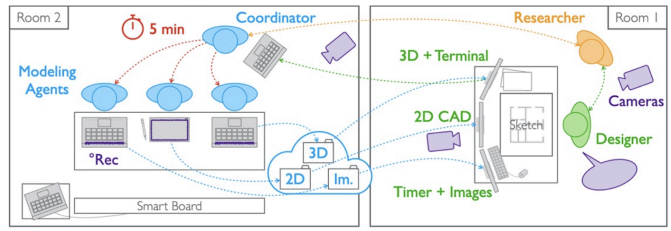


**FIGURE 7:** Experiment principle - a design session instrumented through a Wizard of Oz protocol.

The content generated by the intelligent tool is transmitted to the designer with the objective of capitalizing on the potential of analogical reasoning and stimulating ideation with inspirational stimuli and project rediscovery. Figure 8 illustrates the tool's ability to transform sketches into content that can then be employed in design activities.

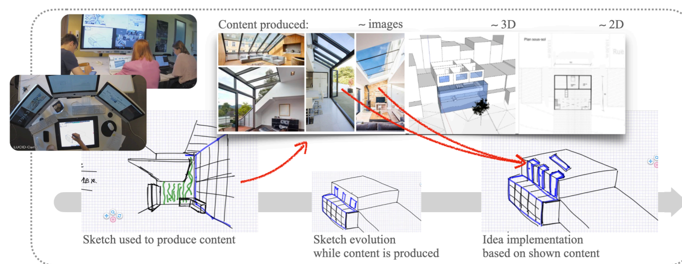


**FIGURE 8:** Example of input from the designer, corresponding output of the pixies and design iteration based on these outputs.

Prior publications from our team have investigated the designer's activity [12,42] and usage of this tool. We have observed a creative exchange between the designer and the tool, which was used as an informative, evaluative, and creative resource. This was achieved through design by analogy and project rediscoveries. Our focus in this paper is thus now the sketch recognition part of the "intelligent tool", through the human agent's sketch recognition activity.

### 3.2 Simulated software's task

In accordance with the Wizard of Oz principle, human agents are placed in the same conditions as the tool. They receive the real-time evolving sketch from the designer in the adjacent room and are informed of the design brief, including the site and program. Each human agent has a specific sub-role. This team includes an "image pixie" who searches for inspiring images online, a "2D pixie" who creates a normalized clean plan of the building to be designed, a "3D pixie" who creates a basic 3D model of the building, and a "coordinator" who manages the team and triggers the sending of the pixies' creations back to the designer every 5 minutes. During this five-minute interval, the designer continues to engage in sketching and design activities. The roles of the three pixies have been selected in accordance with the prevailing practice of utilizing reference images and the capacity of multiple representations (both in 2D and 3D) to facilitate rediscovery.

To carry out their production task and place them in the same conditions as the future technology they are simulating, the modelers have been given specific instructions to follow:

- forbidding them to design an architectural proposal, their role being limited to translating the received representations;
- providing them with a 2D and 3D library of standard furnishings to use by-default;
- specifying the by-default measurements to be assumed, unless otherwise stipulated by the designer, for wall thicknesses, ceiling heights, roof slopes, etc.
- providing them with the site's layout plan and the 3D model;
- informing them of the content of the design brief.

### 3.3 Participants and data collection

Two teams of four pixies, totaling eight subjects, were observed. They were all graduate students in architecture and engineering. The students were selected based on their performance in a preliminary assessment, and they will assume the same role for each design capsule, the one in which they demonstrated the greatest efficiency. The designers (N=9) varied in gender, professional experience, and architectural sensibility, resulting in documentation of the interpretation of nine creatively really distinct projects' propositions.

The teams of pixies are observed during nine one-and-a-half-hour design sessions, resulting in 54 hours of architectural sketch interpretation. The actions of these human agents were recorded through room and screen recordings, allowing for observation of their activities using the AEIOU format to structure field observations [E-LAB, in 43]. We then conduct self-confrontation semi-structured interviews with the pixies. Based on these recordings, we identify questionable and illustrative moments, i.e. every disruptive moment or ones diverging from usual design activities theories. We then discuss these moments with each human agent, showing them the pre-selected video samples and asking them to explain the rationale behind their actions. The starting questions are "How were you able to recognize this part of the sketch? What was your strategy and the elements you used?". We then elaborate on their answers with new questions, individually adapted to the discussion. This approach allows us to access an extensive collection of overall representative behaviors, as well as singular unexpected behaviors and their declared workflow.

### 3.4 Data coding

The interviews are transcribed and coded according to the principles of the Grounded Theory Method, following a method of coding elaborated by Lejeune [44], as shown in Figure 9. Each action noted in the AEIOU Method or explained during the interviews is conceptualized as a 'tag', visually represented by a verb in a box. When saturation is achieved, meaning that all observed or cited actions are represented by a tag, we qualify the articulation between them. These relations can be conjoint (indicated by a green arrow), inverse (red arrow), discrete (square box), continuous (hexagonal box), or dependent on conditions (dotted link showing the condition). As an easily understandable example, extracted from Lejeune's theory [44], the hip-hop dancing practice can be documented as shown in Figure 9. Coding by applying labels to describe what is happening, in its abductive interplay between data and researcher, is recognised by many researchers as a rigorous empirically based approach, if done to saturation, with rigor and to provide greater conceptualisations, that is powerful to study phenomena where there is little research [45].

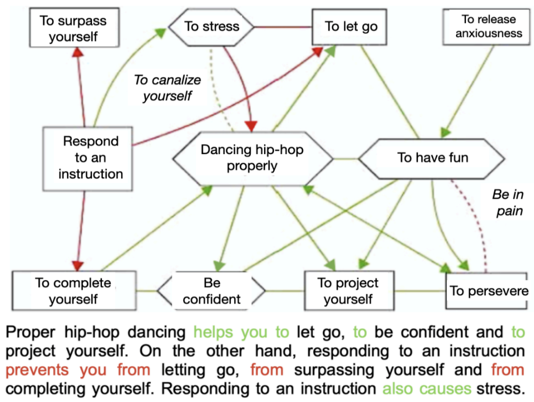


**FIGURE 9:** Example of Grounded Theory Method's schema [44].

## 4. RESULTS AND DISCUSSION

As a reminder, this study aims to understand the challenges of architectural sketch recognition by characterizing the human workflow in performing this task. We examine the resources utilized to understand the architectural object being designed, as well as the procedure constructing a mental model from received graphic features.

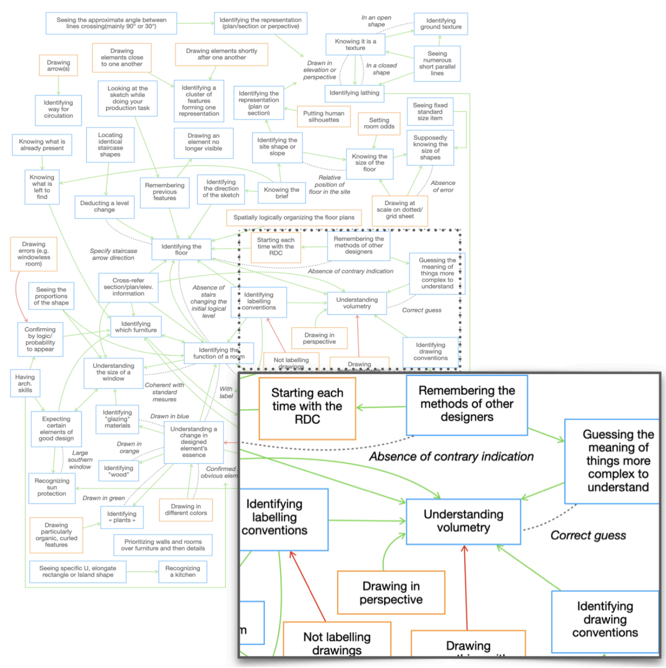


**FIGURE 10:** GTM diagram describing human agents' actions.

Based on the data collected documenting the sketch recognition task done, we obtain a general GTM diagram (Fig. 10), which details all the human agent's actions and sub-actions implemented to carry out the sketch recognition task, as they were either observed on camera or described in interviews.

### 4.1 The need for a functional model

As a first result, we observe that human agents seek to identify the function of the elements sketched, beyond just shapes and symbols (Fig. 11). It appears necessary for them to push their understanding of the sketch to the construction of a **functional mental model** of the designed object, in order to be able to carry out their production tasks. Building a geometric and topological mental model is not enough to collect all the data that will be needed. In addition to information on shapes, zones and symbols, agents need information on function-spaces, furniture and functional characteristics, so they can decide which inspirational images to produce.

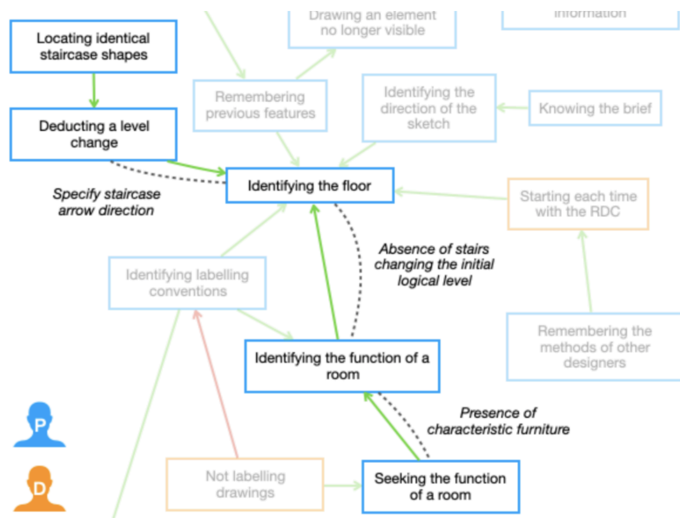


**FIGURE 11:** Seeking for functional information (GTM extract).

### 4.2 A three step sketch recognition process

Once the sketch has been captured, the sketch recognition task leading to the elaboration of the functional mental model of the designed object is in fact a succession of three sub-tasks: synthesis, recognition and interpretation.

#### 4.2.1 Line synthesis

When seeing the sketch, the first action of the agents is to synthesize the lines received. Indeed, architects often multiply the lines symbolizing the same element to reinforce its mass or confirm its location. They also sketch both the alternatives and the final proposal, often on the same drawing, superimposing the different solutions and then anchoring their final choice by passing over its lines several times. It is thus necessary to differentiate between strokes representing architectural elements, graphic construction strokes, annotation strokes, texture strokes and stylistic strokes. As an illustrative example, an agent said that "*He* [the designer] *draws a lot of short parallel lines close together, so we understand that it's a texture of material. A lathing of something. It's very different from the long, rather straight strokes he makes afterwards for walls (...) And then he'll go over the same wall lines several times, so we know there's no new information*".

Agents sort perceived lines into **three families**: lines that mean nothing, lines that embellish and lines that convey information. Only the latter are retained and anchored in the agent's visual memory at the end of this synthesis action.

#### 4.2.2 Line recognition

The agents then recognize these lines as shapes located in relation to each other. They thus construct a topological geometrical mental model of the drawn object. From the lines selected during the synthesis stage, combined with the lines previously seen and stored in their memory, the agents now perceive various closed or open shapes defining spaces with certain adjacent or inclusive relationships between them, as well as symbols An agent said about a drawing: "*it's the rectangular geometric shape formed by four long, more or less straight lines that makes it a room. (...) And here we have a series of parallel or perpendicular lines that form a solid U inside the earlier rectangle. Next, we'll be able to tell that this is kitchen furniture*".

To proceed to the line recognition, the agents also use the memorization of previously seen lines which are no longer necessarily visible, either because they have been erased or because they are in another part of the drawing sheet (Fig. 12). This is essential here to build a mental model of the whole object and not just the part visible at the moment. This also helps build-up a global understanding of the drawing, rather than a collection of snapshots of unrelated parts of the object. This **memory effect** is particularly essential for architectural sketches that develop over several building levels or drawings in plans, sections and elevations.

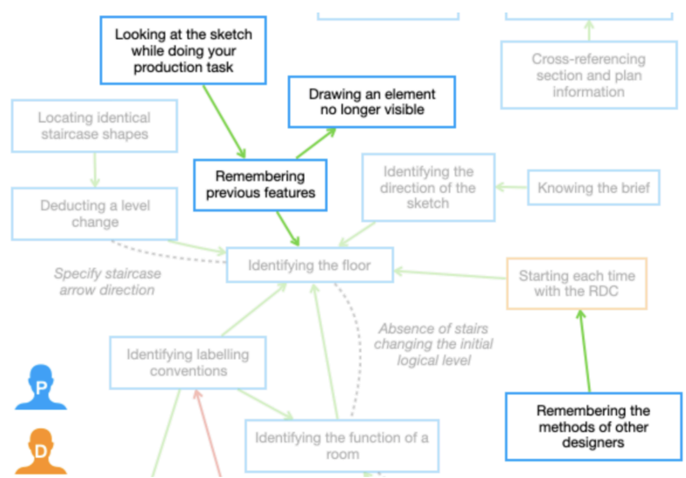


**FIGURE 12:** Using memory to gather information (GTM extract).

#### 4.2.3 Line interpretation

The last sub-action carried out by human agents to move from the topological geometric mental model they have so far constructed to the functional mental model they need is the most complex, and the one that brings out a number of very interesting strategies. To interpret the visible geometry and deduce the functions and characteristics of the shapes, agents call on 4 key resources: (1) knowledge of the context, (2) knowledge of the designer, (3) knowledge of the architecture, and (4) knowledge of the designer's intentions.

**Contextual knowledge** refers to known information about the requested program, the project site and the phase of the design process in which the session takes place. Indeed, knowing the site on which the project is to be built, as well as the program requested, already enables the agents to deduce a number of characteristics expected in the designed project. On a steeply sloping site, you can expect at least two levels (one at garden level, the other at street level). Based on the width of the lot, they can guess whether it's a semi-detached, 3-facade or 4-facade housing project. Knowing that a playroom and an office are required means that these functions can be found in the spaces drawn, even if the layout of the rooms is not shown.

Their **knowledge of the designer** refers to a learning mechanism of the agents. They start to learn the personal codes of representation used by the designer to better understand the sketches. One subject, for example, began to use colors to identify the different types of rooms - bedrooms, bathrooms, corridors, etc. - which the agents then understood (Fig. 13a) and used as information to better understand the project's evolution.

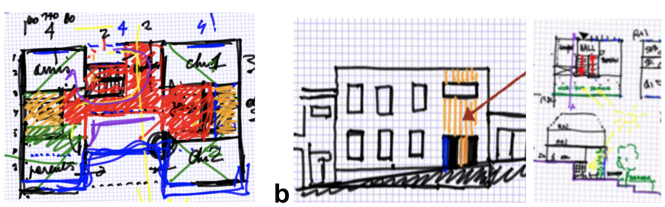
a b

**FIGURE 13: a.** Example of construction of the knowledge of the designer **b.** Color coding in component sketching.

This learning process is as much about absorbing personal drawing codes (colors, symbols, abbreviations, etc.) as it is about design methods (designing plan by plan, progressing in detail, going back and forth between plans and sections, testing different versions with little detail, etc.). The agents also mention a recurrence across all the designers in the color codes used. Walls are represented in dark, cold colors such as blue or black, while the detailed layout is represented in black, blue or orange, and the annotations are made in warm colors (yellow, orange, red, purple) or in black. In addition, blue commonly symbolizes glazing or water; green, vegetation; yellow, light; and orange, wood (Fig. 13b). Black remains the default color.

The use of color codes should be tempered. Although the meaning of a color is consistent and not changed by the designer during its design process, all elements of the same essence are not systematically colored. For example, an element is colored blue in the façade to emphasize its glazed nature as opposed to the solid door, but this does not mean that everything that is not blue is not glazing (Fig. 13b). So color is information, but the absence of color is not.

Their **knowledge of architecture** refers both to knowledge of architectural drawing codes and to the principles of good architectural composition. Architectural codes are, of course, the least ambiguous way of identifying drawn components. For example, a thicker wall is a cut or load-bearing wall, as opposed to a low wall or partition, which is drawn thinner. Doors, staircases, dining tables, beds, sinks, bathtubs and toilets all have their own unique symbolism (Fig. 14). By extrapolating these codes, a room will be a bedroom if it has a bed, a kitchen or dining room if it has a table. Finally, components are also sometimes listed or labeled in the sketch.

But beyond architectural drawing codes, knowledge of the architectural composition and what can be expected in terms of spatial planning can be used to deduce the meaning of uncoded lines. A shape can be understood because it is associated with another, reducing its potential for meaning to a single solution. Let's take a circle as an example: this basic shape can, a priori, symbolize many things, such as an area, a carpet, cooking stoves, a table, a chair, etc. If this circle is intersected by other strokes and is wide, it's more likely to be an annotation delimiting an area. If it's in the middle of a room, it represents a carpet. If this circle, in the middle of a room, is surrounded by smaller identical circles, squares or lines, it symbolizes a table. If, on the other hand, it is grouped with one or three other circles, all inscribed in a rectangle or square, it represents a stove. The scale of the lines also plays a role in interpreting the sketches. Take the same rectangle, thin and long, with its two diagonals marked: this is the architectural code for a tall cabinet. However, if this rectangle takes up a third of a room's surface area, it becomes the cross symbolizing the emptiness of a mezzanine. And this can only be deduced if the agent understands architecture.

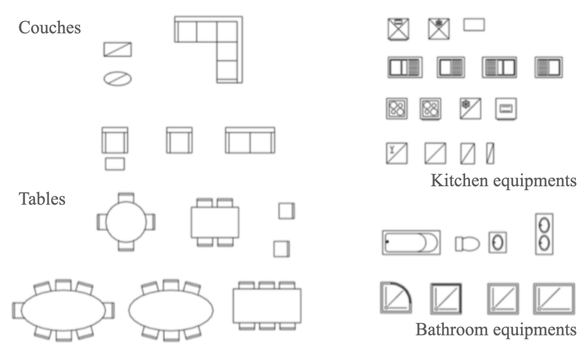


**FIGURE 14:** Example of common furniture symbols and codes.

Finally, their **knowledge of the designer's intentions** is built up as the design session progresses and as the functional mental model of the designed object is constructed. The agents perceive the concepts and principles structuring the proposal that the designer sketches out as he goes along, which helps them deduce where he's going. This progressive iterative understanding of the designer's architectural intentions is also possible thanks to the agents' architectural knowledge.

### 4.3 Holistic human agent's workflow

With a better understanding of the recognition actions, strategies, and resources used, we can summarize the human agents' architectural sketch recognition activity using the following holistic model (Fig. 15).

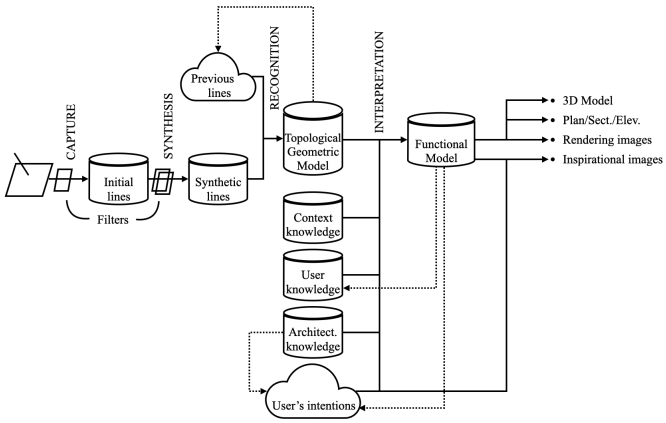


**FIGURE 15:** Holistic model of the sketch recognition process.

This process begins with the reception of the sketch (far left) and progresses through three recognition steps (line synthesis, recognition, and interpretation) before ending with the start of the task of producing the deliverables that the "software" sends back to the designer (far right). This workflow starts with the initial lines being used to create a mental model of the sketch by performing visual filtering to remove unnecessary lines and retaining only the synthetic lines. The features - together with the memory of other previous lines - are recognized as elements (shapes, zones, equipment symbols, etc.). This model is then used to complete the synthetic features received with the previous features. By combining their knowledge of the design context (brief, site), the subject's personal drawing codes (gradually built up), their architectural knowledge (sequences of often adjacent functions, expectation of specific furnishings, etc.), and the designer's architectural intentions (also gradually discovered), they are able to interpret the geometric and topological mental model. This interpretation results in a functional model of the designed building, which identifies the various function-spaces, their boundaries and connections, and the aesthetic/functional characteristics of the furniture elements. The functional model provides feedback on the designer's intentions and personal drawing codes to the agents. The agents then produce inspirational images and other external representations to stimulate the designer's creativity in line with their intentions and the project's direction.

In addition to the discovered resources and strategies, we observed two interesting phenomena : the usefulness of the dynamic evolution of the sketch and the need to make design choices. Indeed, beyond all the possible deduction of the meaning of lines, the key to understanding architectural sketches lies in their **dynamic evolution**. Understanding a sketch taken from its context and frozen at a given moment in time can be extremely complicated. The temporality of the appearance of lines, and the knowledge of the project built up as it is represented, is a crucial key to understanding architectural sketches. Furthermore, for some of the sketches they received, the agents had to **make architectural choices**, despite their instructions to stick to representation, in order to accomplish their task. When designers don't draw to a realistic scale, agents have had to decide between respecting the proportions of the drawing and therefore representing furniture that is larger, or smaller, than normal, or drawing furniture with standard dimensions and therefore not filling the space of the room or overloading it. The boundary is a tricky one to define, as architects are just as likely to design custom-made furniture with specific intentions to create a spatial effect, as they are to furnish their rooms with standardized furniture. How, then, to distinguish between a scale intentionally different from standard dimensions and a representational error? Faced with this difficulty of positioning, the agents were asked to represent the furniture in standard dimensions, waiting to be corrected if necessary. Some designers thus became aware of their dimensioning errors thanks to the "software".

### 4.4 Insights for future intelligent design tool

While some of the boxes shown in the holistic model in Figure 15 can be easily replaced by currently existing techniques, others present real challenges. The first main challenge lies in the initial step of **synthesizing the received lines**. As we have seen, sketches can consist of numerous lines, some of which may carry implicit or explicit information, while others may be texture or unnecessary lines that obscure the drawing's legibility. Additionally, **integrating information from multiple parallel sketches** (such as plans, cross-sections, and detailed sub-sketches) is also a challenge. After that, transitioning from synthetic lines to a topological geometric model is a technique that has already been mastered in many design domains [19-21], including architecture [28, 29].
Interpreting drawings by recognizing drawing codes and symbols is also a well-established practice [30, 31]. But the limits are reached when users add personal codes or do not use the pre-recorded codes. Indeed, while using architectural drawing code databases and pre-encoded context information is easily done by software agents, the challenge is to utilize the designer's habits and intentions, and to predict the probability of certain design elements based on the project context or rules of good composition. **Populating the databases for user knowledge, architectural knowledge, and design intentions** in this holistic model will be a substantial task.
Finally, despite being already possible to create a model for functional understanding of a drawing from simple sketches in design domains with explicit and objectifiable codification [35, 39], it remains very challenging for architectural sketches, which are inherently complex, incomplete, and contain implicit information.

Based on the priorly presented results, we provide insights for overcoming these challenges and developing powerful sketch-based generative AI tools for complex design situations where current tools are insufficient.
Firstly, the sketch recognition module should be **integrated into the drawing medium** instead of relying on frozen images or sketch extracts. This allows access to sketches under construction, providing more information such as the temporality of line appearance and process perception beyond what is currently visible. The tool should also have the ability to **memorize and store features** that have been seen, in addition to those currently visible at the time of the recognition request. This helps the line recognition and the building of a

global understanding of the drawing, rather than a collection of snapshots of unrelated parts of the object. This is particularly important for sketches that develop over several building levels or drawings in plans, sections, and elevations.

Secondly, the tool should **combine symbolic and connectionist logic**. Symbolic logic uses predefined rules and explicit instructions to narrow down the field of possible interpretation. On the other hand, connectionist logic relies on statistical recognition probabilities and the knowledge provided to solve the problem. The resources used by human agents to understand the received drawing belong to both logics. The recognition strategy involves deducing the meaning of features based on shape associations, feature scale, color codes, architectural codes and the probability of expecting a particular compositional element (according to rules of good architectural composition principles). Another part of the recognition strategy involves learning the designer's habits and intentions to start recognizing his/her own drawing codes and the probability of expecting a particular element according to his/her recurrent design method and even architectural style.

Finally, in order to be able to generate appropriate inspirational images for the object being designed, the tool, in recognizing the sketch, must go as far as a fine level of identification of the functions (circulation space, kitchen, living room, etc.) and of the architectural or aesthetic characteristics (daylight, massiveness, lightness, natural materials, etc.) of the various elements drawn. This has been shown by the need to build a **functional mental model** to complement the topological geometric model.

## 5. CONCLUSION

In this work, a wizard of oz protocol for an architectural design task was used in order to surface future requirements for sketch-based generative AI systems that could be used in design practice. To answer our research questions, we have highlighted a three-step human recognition activity - synthesis, recognition and interpretation - that involves the mobilization of four knowledge resources - related to the project context, the design domain, the designer's habits and the designer's intentions - and is enabled by two key characteristics: visual memory and the dynamic nature of the sketches received in this experience. Studying this recognition activity highlighted the specific challenges of understanding complex design sketches and provided insights for designing AI tool workflows and overcoming the capability gap of current systems.

Based on our results, we find that future sketch-based generative AI tools should incorporate: (1) integration in the dynamic construction of the sketch and continuous storage of features in memory; (2) synthesis of symbolic and connectionist logic to operate various recognition resources; and (3) extension beyond geometrical models to build a functional model of the object, in order to be able to generate interesting and accurate inspirational images. Taken together, these findings can be incorporated into the development of new approaches to recognize sketches at the fundamental level, and a perspective to recognize sketches that were previously too complex at the applied level. Finally, they inform the inclusion of new resources and software architecture within AI tools.

## REFERENCES


[1] Roberts, M., Allen, S., & Coley, D. (2020). Life cycle assessment in the building design process–A systematic literature review. *Building and Environment*, 185, 107274

[2] Leclercq, P., & Heylighen, A. (2002). *5,8 Analogies per hour. A designer's view on analogical reasoning*. Paper presented at the AID'02 Artificial intelligence in design.

[3] Goel, A. K. (1997). Design, analogy, and creativity. *IEEE expert, 12*(3), 62-70.

[4] Linsey, J. S. (2007). *Design-by-analogy and representation in innovative engineering concept generation*. The University of Texas at Austin.

[5] Sio, U. N., Kotovsky, K. & Cagan, J. (2015). Fixation or inspiration? A meta-analytic review of the role of examples on design processes. *Des. Stud. 39*, 70–99.

[6] Jiang, S., Hu, J., Wood, K. L., & Luo, J. (2022). Data-driven design-by-analogy: state-of-the-art and future directions. *Journal of Mechanical Design, 144*(2), 020801.

[7] Elara, M. R. (2023). Do Analogies and Analogical Distance Influence Ideation Outcomes in Engineering Design? V. Srinivasan, Binyang Song, Jianxi Luo, Karupppasamy Subburaj, Mohan Rajesh Elara, Lucienne Blessing, and Kristin Wood. *Applied Cognitive Science and Technology: Implications of Interactions Between Human Cognition and Technology*, 211.

[8] Ball, L. J., & Christensen B. T. (2019). Advancing an understanding of design cognition and design metacognition: Progress and prospects. *Design Studies 65*, 35-59.

[9] Jaruga-Rozdolska, A. (2022). Artificial intelligence as part of future practices in the architect's work: MidJourney generative tool as part of a process of creating an architectural form. *Architectus 3*(71), 95-104.

[10] Paananen, V., Oppenlaender J., & Visuri A. (2023). Using Text-to-Image Generation for Architectural Design Ideation. Preprint.

[11] Safin, S., Maitrallin, M., Fruchard, B., & Lecolinet, E. (2021). Appropriation and memorisation processes of gestural shortcuts on trackpad: Longitudinal study of users' strategies and impact of visuo-semantic aid. In Actes de la 32e conférence francophone sur l'Interaction Humain-Machine (IHM'20.21) (pp. 14-1). ACM.

[12] Baudoux, G, & Goucher-Lambert, K. (in press). Automating Analogical Reasoning: A Wizard of Oz study on the benefits and pitfalls of a sketch-based AI image generator for design. DCC International Conference.

[13] Baudoux, G. (2024). The Benefits and Challenges of AI Image Generators for Architectural Ideation: Study of an alternative human-machine co-creation exchange based on sketch recognition. *International Journal of Architectural Computing*.

[14] Stahovich, T. F. (1996). *SketchIT: A sketch interpretation tool for conceptual mechanical design.* USA: PhD thesis.

[15] Davis, R. (2002). Sketch understanding in design: Overview of work at the MIT AI lab. In *Sketch Understanding, Papers from the 2002 AAAI Spring Symposium* (pp. 24-31).

[16] Notowidigdo, M. J. (2004). *User-directed sketch interpretation* (Doctoral dissertation, Massachusetts Institute of Technology).
[17] Ding, C., & Liu, L. (2016). A survey of sketch based modeling systems. *Frontiers of Computer Science*, *10*, 985-999.
[18] Zhang Z, Jin Y (2022). Data-enabled sketch search and retrieval for visual design stimuli generation. Artificial Intelligence for Engineering Design, Analysis and Manufacturing 36, e25, 1–18
[19] Zhang, Z., & Jin, Y. (2021). Toward Computer Aided Visual Analogy Support (CAVAS): Augment Designers Through Deep Learning. *Proceedings of the ASME 2021 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Virtual, Online, Aug. 17–19, p. V006T06A057.
[20] Kim, J., & Maher, M. L. (2023). The Effect of AI-Based Inspiration on Human Design Ideation. *Int. J. Des. Creativity Innov., 11*(2), pp. 81–98.
[21] Arora R, Darolia I, Namboodiri VP, Singh K & Bousseau A (2017). Sketchsoup: exploratory ideation using design sketches. *Computer Graphics Forum 36*, 302–312.
[22] Kazi, R. H., Grossman, T., Cheong, H., Hashemi, A., & Fitzmaurice, G. (2017). DreamSketch: Early stage 3D design explorations with sketching and generative design. In *Proc. Of the 30th annual ACM symposium on user interface software and technology*, UIST '17 (pp. 401e414). New York, NY, USA: Association for Computing Machinery.
[23] Jiang, S., Luo, J., Ruiz-Pava, G., Hu, J., & Magee, C. L. (2021). Deriving Design Feature Vectors for Patent Images Using Convolutional Neural Networks. *J. Mech. Des., 143*(6), p. 061405.
[24] Kwon E, Pehlken A, Thoben KD, Bazylak A & Shu LH (2019). Visual similarity to aid alternative-use concept generation for retired wind-turbine blades. *Journal of Mechanical Design 141*, 031106
[25] Kwon, E., Huang, F., & Goucher-Lambert, K. (2022). Enabling Multi-Modal Search for Inspirational Design Stimuli Using Deep Learning. *Artif. Intell. Eng. Des. Anal. Manuf., 36*(1), p. e22.
[26] Valveny, E., & Martí, E. (2000). Deformable Template Matching within a Bayesian Framework for Hand-Written Graphic Symbol Recognition. In: Chhabra, A.K., Dori, D. (eds) Graphics Recognition Recent Advances. GREC 1999. *Lecture Notes in Computer Science*, vol 1941. Springer, Berlin, Heidelberg.
[27] Lee, S., Feng, D., Grimm, C. & Gooch, B. (2008), A Sketch-Based User Interface for Reconstructing Architectural Drawings. *Computer Graphics Forum, 27*: 81-90.
[28] Johnson, G., Gross, M., Yi-Luen Do, E., & Hong, J. (2012). Sketch it, make it: sketching precise drawings for laser cutting. In *CHI'12 Extended Abstracts on Human Factors in Computing Systems*. ACM, 1079–1082
[29] Hsiao, C.-P.. (2015). Solidsketch: Toward enactive interactions for semantic model creation. In P*roceedings of the 2015 ACM SIGCHI Conference on Creativity and Cognition*. ACM, 329–330.
[30] Juchmes, R., Leclercq, P., & Azar, S. (2004). *A multi-Agent System for Architectural Sketches Interpretation.* Paper presented at the Eurographics Workshop on Sketch-Based Interfaces and Modeling.
[31] Demaret, J.-N., & Leclercq, P. (2012). Génération automatique d'un modèle de bâtiment à partir d'un croquis. Actes du 5ème séminaire de Conception Architecturale Numérique (SCAN'12)
[32] Wang, X., Chen, X., & Zha, Z. (2018). Sketchpointnet: A compact network for robust sketch recognition. In *2018 25th IEEE International Conference on Image Processing (ICIP)* (pp. 2994-2998). IEEE.
[33] Liu, F., Deng, X., Lai, Y. K., Liu, Y. J., Ma, C., & Wang, H. (2019). Sketchgan: Joint sketch completion and recognition with generative adversarial network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 5830-5839).
[34] Gross, M. and Yi-Luen Do, E. (1996). The electronic cocktail napkin. *Design studies 17*(1), 53–69.
[35] Davis, N., Hsiao,, C.-P. Yashraj Singh, K., and Magerko, B. (2016). Co-creative drawing agent with object recognition. In. *Twelfth ArtificialIntelligence and Interactive Digital Entertainment Conference*.
[36] Lucas, P., & Martinho, C. (2017). Stay Awhile and Listen to 3Buddy, a Co-creative Level Design Support Tool. In *ICCC*, (pp. 205-212.
[37] Liapis, A., Yannakakis, G., & Togelius, J.. (2013). Sentient Sketchbook: Computer-aided game level authoring.. In *FDG*. 213–220.
[38] Steinfeld, K. (2020). Sketch2pix: Machine-Augmented Architectural Sketching. In *Computational Design and Robotic Fabrication Conference*.
[39] Karimi, P., Rezwana, J., Siddiqui, S., Maher, M. L., & Dehbozorgi, N. (2020). Creative sketching partner: an analysis of human-AI co-creativity. In *Proceedings of the 25th International Conference on Intelligent User Interfaces* (pp. 221-230).
[40] Brie, P., Burny, N., Sluÿters, A., & Vanderdonckt, J. (2023). Evaluating a Large Language Model on Searching for GUI Layouts. *Proceedings of the ACM on Human-Computer Interaction*, *7*(EICS), 1-37.
[41] Rietz, F., Sutherland, A., Bensch, S., Wermter, S., & Hellström, T. (2021). WoZ4U: An Open-Source Wizard-of-Oz Interface for Easy, Efficient and Robust HRI Experiments. *Frontiers in Robotics and AI*, *8*.
[42] Baudoux G., Safin, S. (In submission). Study of the multi-instrumented reflective conversation activity in preliminary design. In *Design Sciences*.
[43] Wasson, C. (2000). Ethnography in the field of design. *Human organization, 59*(4), 377-388.
[44] Lejeune, C. (2019). Manuel d'analyse qualitative. De Boeck Supérieur.
[45] Cole, T., & Gillies, M. (2022, April). More than a bit of coding:(un-) Grounded (non-) Theory in HCI. In CHI Conference on Human Factors in Computing Systems Extended Abstracts (pp. 1-11).

 