

**SPACESHEETS:
DESIGN EXPERIMENTATION IN LATENT SPACE**

BY BRYAN JUN YAN LOH

A thesis submitted to Victoria University of Wellington in fulfilment
of the requirements for the degree of Master in Design Innovation

Victoria University of Wellington

2018

Thanks to Tom White, for the insightful and
generous supervision of this research.

To the creators of the knowledge from which
this research draws and builds upon.

To the staff at Victoria University involved in
organising the research programme.

To my friends and family.

ABSTRACT

Computational design tools enable designers to construct and manipulate representations of design artifacts to arrive at a solution. However, the constraints of deterministic programming impose a high cost of tedium and inflexibility to exploring design alternatives through these models. They require designers to express high-level design intent through sequences of low-level operations. Generative neural networks are able to construct generalised models of images which capture principles implicit within them. The latent spaces of these models can be sampled to create novel images and to perform semantic operations. This presents the opportunity for more meaningful and efficient design experimentation, where designers are able to express design intent through principles inferred by the model, instead of sequences of low-level operations.

A general purpose software prototype has been devised and evaluated to investigate the affordances of such a tool. This software — termed a *SpaceSheet* — takes the form of a spreadsheet interface and enables users to explore a latent space of fonts. User testing and observation of task-based evaluations revealed that the tool enabled a novel top-down approach to design experimentation. This mode of working required a new set of skills for users to derive meaning and navigate within the model effectively. Despite this, a rudimentary understanding was observed to be sufficient to enable designers and non-designers alike to explore design possibilities more effectively.

TABLE OF CONTENTS

List of Figures	1
List of Tables	5
Introduction	7
Methodology	11
Background	19
Hypothesis	31
Design Proposition	33
Design Process	37
Results	77
Conclusion	81
Bibliography	85

LIST OF FIGURES

LIST OF FIGURES

Figure 1: A two-dimensional space of wine glasses. Operations are performed to transform its form. Source: Drawing by Patrick Hebron (2017). <i>Rethinking Design Tools in the Age of Machine Learning</i> . Retrieved from URL: https://medium.com/artists-and-machine-intelligence/rethinking-design-tools-in-the-age-of-machine-learning-369f3f07ab6c	26
Figure 2: Diagram of a latent space of faces. Points within the latent space can be decoded into observable images of faces. Source: Image by Tom White (2017). <i>Sampling Generative Networks</i> . Retrieved from URL: http://arxiv.org/abs/1609.04468	27
Figure 3: Naive bold comparison with actual bolding. Source: Drawing by Shan Carter & Michael Nielsen (2017). <i>Using Artificial Intelligence to Augment Human Intelligence</i> . Retrieved from URL: https://distill.pub/2017/aia/	28
Figure 4: An interpolation between two faces. Source: Image by Tom White (2017). <i>Sampling Generative Networks</i> . Retrieved from URL: http://arxiv.org/abs/1609.04468	29
Figure 5: Applying a “smile vector” to a face. Source: Image by Tom White (2017). <i>Sampling Generative Networks</i> . Retrieved from URL: http://arxiv.org/abs/1609.04468	29
Figure 6: Spreadsheet software being used to create a cost estimation. Source: Wikipedia. A cost estimate spreadsheet prepared with computer spreadsheet software (2011). Retrieved from URL: https://commons.wikimedia.org/wiki/File:Estimating_Spreadsheet.png	35
Figure 7: An interpolation between two glyphs in the DeepFonts model.	39
Figure 8: Diagram of the SpaceSheet. The Data Picker (front left) is used to populate cells in the spreadsheet (front right) with variables from latent space (back). The spreadsheet can be used to define operations to arrive at new points in latent space.. . . .	41

SPACESHEETS: DESIGN EXPERIMENTATION IN LATENT SPACE

Figure 9: Screenshot of the Colour Spreadsheet.....	44
Figure 10: A series of interpolations between two colours with the slider element.....	44
Figure 11: Screenshot of Version One.....	46
Figure 12: Cell references in cells being edited are highlighted.....	47
Figure 13: Controlling the RANDFONT cell with a slider element.	48
Figure 14: The RANDFONT cell can be randomised by clicking a button.	48
Figure 15: A selection of Data Picker experiments.....	49
Figure 16: Intermediate variable interpolations as the Data Picker is zoomed in upon.	50
Figure 17: Tweening between two glyphs using a slider element.	51
Figure 18: Isolating and applying a bold vector in the SpaceSheet.....	52
Figure 19: Illegible representation of an attribute vector.	54
Figure 20: John's decision tree layout.	55
Figure 21: Operations are suggested based on the selection.	58
Figure 22: Suggested cells to be populated are highlighted as the user hovers over a suggested operation.....	59
Figure 23: Variety of Double SmartFill configurations.....	59
Figure 24: Group SmartFill.....	60
Figure 25: Miscellaneous Corner SmartFills.	60

LIST OF FIGURES

Figure 26: A grid of interpolations created with the Group SmartFill.	61
Figure 27: Font drawer with stored samples from the spreadsheet.	61
Figure 28: Information overlay on the Data Picker illustrating the viewport position and scale.	62
Figure 29: Data Picker with multiple layouts and a UI overlay.	63
Figure 30: Similar fonts with minimal differences have been selected for the Standard layout.	63
Figure 31: Data Picker of face drawings.	65
Figure 32: Demo sheet which isolates and applies attribute vectors to a variable.	69
Figure 33: Cell references within the selected cell are highlighted.	71
Figure 34: Ringo's "crossword puzzle" layout.	73

SPACESHEETS: DESIGN EXPERIMENTATION IN LATENT SPACE

LIST OF TABLES

LIST OF TABLES

Table 1: Creativity Support Index Agreement Statements.	16
Table 2: Summary of Research Methods.	17
Table 3: John and George's Creativity Support Index Agreement Scores.	56
Table 4: Paul's Creativity Support Index Agreement Scores.	67
Table 5: Ringo's Creativity Support Index Agreement Scores.	74
Table 6: Creativity Support Index Agreement Scores from all User Testing Sessions.	77

SPACESHEETS: DESIGN EXPERIMENTATION IN LATENT SPACE

INTRODUCTION

INTRODUCTION

Computational design tools enable designers to construct and manipulate models — representations of design artifacts — to arrive at a solution. However, the constraints of deterministic programming impose a high cost of tedium and inflexibility to exploring design alternatives through these models. They require designers to express high-level design intent through sequences of low-level operations.

Generative neural networks are able to construct generalised models of images which capture principles implicit within them. The latent spaces of these models can be sampled to create novel images and to perform semantic operations.

This presents the opportunity for more meaningful and efficient design experimentation, where designers are able to express design intent through principles inferred by the model, instead of sequences of low-level operations.

The research investigates the affordances and potential of such a tool through the development and evaluation of *SpaceSheets*, a general purpose spreadsheet software prototype which enables users to sample latent spaces.

This thesis documents the process of designing the tool and the results and conclusions gathered from its evaluation. To develop a more complete understanding of the project, it is highly recommended that the reader has firsthand experience with the tool.

The software along with an accompaniment of short tutorials are available at this URL:

<https://gitlab.com/bryanlohjy/spacesheets-mdi-eval>

If the project does not exist at that URL, please request the digital files submitted alongside this thesis. Included in this set of files is a copy of the software, along with instructions on how to get started.

SPACESHEETS: DESIGN EXPERIMENTATION IN LATENT SPACE

AIMS AND OBJECTIVES

The research aims to explore the potential of latent spaces to be used as a tool for design experimentation. To achieve this, the following research objectives have been developed:

1. Situate the research within the context of design experimentation, design tools and the state of generative models.
2. Establish an understanding of the properties of tools which support design experimentation.
3. Devise a solution and evaluate it.
4. Derive conclusions.

DEFINITION OF KEY TERMS

Design

Used as a verb, this is the act of conceiving a set of specifications for a solution to a problem. When used as a noun, “*design*” refers to the solution conceived by a design process.

Computational Design Tools

“*Computational Design Tools*” refers to software which is used to execute design intent in a design process. This is commonly abbreviated to “*Design Tools*” when used in the thesis.

High-level vs. Low-level

“*Level*” refers to the level of abstraction of a given object or action. High-level entities are more abstract and refer to concepts and ideas. Low-level entities are less abstract and refer to details and specifics.

INTRODUCTION

This is often used in the thesis to explain the differences between abstract, high-level design intent (e.g. “*More elegant*”) and concrete, low-level operations in design tools (e.g. “*Move this rectangle 20px to the left*”).

Model

A representation, or description of a system. When used in this thesis, it either refers to a model which describes a range of design possibilities. These are either constructed by the user in design software, or through generative neural networks.

Generative Neural Networks

“*Neural network*” in the thesis refers to artificial neural networks. Artificial neural networks are algorithms which learn to approximate a function to transform an input to an output. This function is learned through a *training step* where it is provided a set of inputs along with its expected outputs. This function is encoded within the network as a set of parameters. These parameters are unintelligible to humans. Due to this, neural networks are described to be a “*black box*”.

Generative neural networks or *Generative neural network models* are algorithms which learn to construct a generalised model from a set of training data. This is achieved by the training its neural networks to learn transformation functions on a smaller set of parameters than that within the training data. By learning how to represent the training data in a more compact form, the model is forced to learn more abstract, high-level representations of the data.

Unless mentioned otherwise in the thesis, these models are trained to generate images.

Latent Variables and Latent Space

Latent variables are the high-level representations learned by the generative neural network models. These variables exist in latent space and can be used to sample the generative model. They are unobservable and have to be decoded through the transformation functions learned by the neural network into observable values. Conversely, observable values can be encoded into latent variables to be operated on in latent space.

SPACESHEETS: DESIGN EXPERIMENTATION IN LATENT SPACE

METHODOLOGY

METHODOLOGY

Research methods are the means of inquiry in a research problem. There are a variety of different research methods which yield different results, making them useful for different situations. Therefore, the selection of appropriate research methods is a key factor in determining the quality and nature of the outcomes of the research.

To determine the appropriate research methods, the research problem should be examined. By examining the goals and nature of the problem, we can posit the appropriate methodology and methods to obtain the necessary data to answer the problem.

This chapter examines the research problem to build an understanding of the nature of the problem. This understanding is then used to determine the type of information necessary to answer the research problem and the research methods which will be used to obtain it.

THE NATURE OF THE PROBLEM

The research aims to generate an understanding of the affordances of latent space exploration as a tool for design experimentation. Stated as a problem, it could be phrased along the lines of: *“In what ways can latent space exploration support design experimentation?”*.

The use of purely deductive research methods to answer this problem is insufficient to arrive at any conclusive evidence about latent space exploration as a design tool. This is due to the fact that the relationship between designers and their tools - user behaviours, workflows, opinions - cannot be captured and modelled specifically for the tool in question. Shneiderman shares the same view, when explaining the difficulty of evaluating Creativity Support Tools, *“...the complex nature of human discovery and innovation cannot be studied like pendulums or solid-state materials”* (Shneiderman, 2007, p. 27).

SPACESHEETS: DESIGN EXPERIMENTATION IN LATENT SPACE

To generate valuable insights, a form should be instantiated, placed in context and evaluated through user observation. In the case of this research, this form consists of the net affordances of its user interface and its underlying system. The context consists of the behaviour and processes of people who engage in design experimentation.

The research reveals itself to be what Rittel & Webber (1973, p. 162) describe as a “*wicked problem*”, where “*the process of solving the problem is identical with the process of understanding its nature*”.

The form of the tool can be seen as a solution instantiated to gather information about the understanding of the problem. As more information emerges from evaluating the solution, the understanding of the problem changes. This new understanding of the problem, in turn, implies a modification of the existing solution.

The identical process in understanding and solving the problem implies that there is no stopping rule in solving the problem. The problem could always be understood better, and so there is an endless amount of possible solutions (Rittel & Webber, 1973). This indefinite cycle can be bounded by engaging in a design process which focuses on discovering the solution to the problem, while it simultaneously builds the understanding of the problem. In *Designerly Ways of Knowing*, (Cross, 1982, p. 7) notes that “*it will always be possible to go on analysing ‘the problem’...It is only in terms of a conjectured solution that the problem can be contained within manageable bounds*”.

RESEARCH THROUGH DESIGN

A solution-focussed approach to the research can be viewed as a design problem, where the goal is to achieve a good fit between the form of the tool and its context. Christopher Alexander describes this fit as a “*relation of mutual acceptability*” between the form and its context and as the goal of all design problems (Alexander, 1964, p. 19).

He elaborates on the idea of fit, describing a good fit as placing both the form and context in “*effortless contact or frictionless coexistence*” (Alexander, 1964, p. 19) with each other.

METHODOLOGY

This fit has to be achieved through exploration. Since the solution and understanding of the problem are co-dependent, the specifications for good fit are not known prior. This ambiguity is approached in a design process with what is described to be “*an initial state of intentional ignorance*”, where the designer is “*completely open to what is emergent in the moment*” in *The Design Way* (Nelson & Stolterman, 2012, p. 44).

In the experiences of researchers in the field of Creativity Support Tools, this approach manifests itself as an iterative process of development, in which they are “*constantly critiquing, adjusting, modifying and revising*” functional prototypes (Resnick et al., 2005). It is through this critical pursuit of good fit where an improved understanding of the problem is generated. There is implicit knowledge in the design of the prototype, making it a “*potential generator of knowledge*” (Stappers, 2007, p. 87). The designing of this prototype is the essence of “*research through design*” (Stappers, 2007, p. 87).

RESEARCH METHODS

The course of the research will be based on a set of development phases identified by Hewett et al. (2005) to develop Creativity Support Tools:

1. Observe activities and problems.
2. Gather user requirements.
3. Design and implement.
4. Evaluate and iterate (repeated as often as needed / possible).
5. Follow up in field longitudinally, over an extended period of time.

The last phase of the process will be omitted to fit the time constraints of the research.

SPACESHEETS: DESIGN EXPERIMENTATION IN LATENT SPACE

Research methods have chosen to best understand the strengths and limitations of the tool. This means qualitative methods which capture user opinions on the tool. This is inspired by the ethos of *Design Ethnography*, to “look deeper into what people do, what tools they use, and how they think” (Salvador, Bell, & Anderson, 2010, p. 35).

The remainder of this section presents the research methods that will be used within each development phase.

Observe activities and problems

A literature review on design experimentation, design tools and sampling latent spaces will be conducted to understand user needs and problems, and the possibilities.

Semi-structured interviews with design experts will be conducted to understand how design tools are used to provide in-depth information on user needs and problems.

Gather user requirements

User requirements will be derived from the understanding developed in the observation phase, along with a review of literature to understand how tools can support design experimentation.

Design and implement

The design will be developed from the user requirements, understanding of the problem and an understanding of the technology available.

Evaluate and iterate

The tool will be evaluated through the user testing procedure as detailed below. The *Thinking Aloud Method* (Lewis & Rieman, 1993, p. 83) will be used in the user exploration and task-based analysis section to capture the user’s thoughts on the effects of specific features of the tool as they arise.

METHODOLOGY

User Exploration

The user is presented with the tool and encouraged to learn how to use it through exploration.

Task-based Analysis:

Search tasks are given to the user to gain an understanding of the different ways a user might attempt to solve a given task. Two types of search tasks are given:

a. Known target search task:

Three images of outcomes created with the tool are presented to the user. The user is then prompted to replicate the results in the image using the tool. A loose time limit of 3 - 5 minutes is given.

b. Ill-defined search task:

The user is tasked to create an outcome for an abstract problem — e.g. “Fun and trustworthy brand”. This task is used to understand how the tool might be used to make design judgements. No time limit is given for this task.

Creativity Support Index Agreement Questionnaire

The *Creativity Support Index* (CSI) (Cherry & Latulipe, 2014) is a psychometric survey which measures six dimensions of creativity support - *Collaboration*, *Enjoyment*, *Exploration*, *Expressiveness*, *Immersion*, *Results Worth Effort* to evaluate the “*ability of a creativity support tool to assist a user engaged in creative work*” (Cherry & Latulipe, 2014, p. 21). It is administered through an agreement statement section and a paired-factor analysis. The resulting data is then used to calculate the index score.

In this research, the CSI Agreement Questionnaire will be administered during the user testing session to evaluate the tool. This questionnaire consists of two agreement statements for each dimension being evaluated — the user would fill in a score between 0 -10 on how much they agree with a statement. The *Collaboration* dimension will be omitted as it is not applicable to the research.

SPACESHEETS: DESIGN EXPERIMENTATION IN LATENT SPACE

This questionnaire will enable the tool to be evaluated with more granularity, enabling more specific feedback on what is working well, and what could be improved on.

The questionnaire is as follows:

Dimension	Agreement Statement
Enjoyment	I would be happy to use this tool on a regular basis.
	I enjoyed using the tool.
Exploration	It was easy for me to explore many different ideas, options, designs, or outcomes, using this tool.
	The tool was helpful in allowing me to track different ideas, outcomes, or possibilities.
Expressiveness	I was able to be very creative while doing the activity inside this tool.
	The tool allowed me to be very expressive.
Immersion	My attention was fully tuned to the activity, and I forgot about the tool that I was using.
	I became so absorbed in the activity that I forgot about the tool that I was using.
Results Worth Effort	I was satisfied with what I got out of the tool.
	What I was able to produce was worth the effort I had to exert to produce it.

Table 1: Creativity Support Index Agreement Statements.

Semi-structured feedback interview

The CSI Agreement Questionnaire will be used as a starting point to encourage thoughtful user feedback on the tool through a semi-structured interview. This interview will be used to understand what was useful, and what wasn't useful to the user in achieving the tasks.

The data collected will be interpreted to provide actionable design improvements for further iterations.

METHODOLOGY

Summary of Research Process

Below is a summary of the research and evaluation methods selected, grouped within the research phases identified in the Creativity Support Tool development framework.

Research Phase	Methods Used
Observe activities and problems	<ul style="list-style-type: none">• Literature review• Semi-structured interviews
Gather user requirements	<ul style="list-style-type: none">• Literature review
Design and implement	<ul style="list-style-type: none">• Designed from the understanding of the problem, and existing feedback
Evaluate and iterate	<ul style="list-style-type: none">• User testing<ul style="list-style-type: none">• Thinking Aloud method• Task-based analysis<ul style="list-style-type: none">• Known target search• Ill-defined search• Creativity Support Index Agreement Questionnaire• Semi-structured feedback interview

Table 2: Summary of Research Methods.

SPACESHEETS: DESIGN EXPERIMENTATION IN LATENT SPACE

BACKGROUND

BACKGROUND

EXPERIMENTATION IN DESIGN

Design involves the generation of a set of specifications of an artifact to solve a goal. The structure of this activity is understood through two contrasting paradigms: *Rational Problem Solving* and *Reflective Practice* (Dorst & Dijkhuis, 1995). Despite their differences, experimentation is essential to arrive at a solution in both paradigms.

In the view of *Rational Problem Solving*, design is described as a search problem within a space for a solution. Newell and Simon (1972) describe this process as “an odyssey through the problem space, from one knowledge state to another, until his current knowledge state includes the problem solution — that is, until he knows the answer” (Newell & Simon, 1972, p. 151).

Within this space, the distinction is made between *nodes* and *operators*. Nodes are described as states of knowledge in the problem space, while operators are relationships between nodes which can be applied to arrive at other nodes in the space. (Newell & Simon, 1972).

The space is structured and not randomly organised. This structure enables the problem solver to search selectively by predicting properties of unsearched parts of the space from information gathered from parts of the space which have been searched (Newell & Simon, 1972). These selective searches are experiments consisting of operator and node choices, evaluated by their efficacy in achieving progress towards a solution.

Simon and Newell (1972) identify two heuristics for evaluating progress towards a solution: *hill climbing* and *means-end analysis*.

Hill climbing is an analogy derived from the heuristic used when climbing up a hill — “If a particular spot is higher, reaching it probably represents progress toward the top” (Newell & Simon, 1972, p. 152). Progress towards a solution is evaluated by the promise of a visited node as a starting point for further searches.

SPACESHEETS: DESIGN EXPERIMENTATION IN LATENT SPACE

Means-end analysis evaluates progress as the difference between the current state of affairs and the desired state of affairs (Newell & Simon, 1972). Progress towards solving the problem is made when the problem solver arrives at a node which differs less from the goal state than the nodes visited previously. This difference serves as the criteria for selecting an operator which minimises a particular difference between the current node and the goal.

The paradigm of *Reflective Practice* (Schon, 1984) describes design as a “*reflective conversation with the situation*”, where the designer “*spirals through stages of appreciation, action and reappreciation*” (Schon, 1984, p. 132). Problems are “*framed*” by the designer’s appreciation of a situation. These problems are then solved through actions which attempt to improve the current situation. The designer’s actions produce unintended changes which require a reappreciation and reframing of the problem (Schon, 1984, p. 131-132).

These actions are experiments to discover what “*consequences and implications can be made to follow from them*”, (Schon, 1984, p. 131) in an attempt to fit the situation to the framing of the problem.

Schon (1984) distinguishes three types of experiments conducted by designers: *exploratory*, *move-testing* and *hypothesis testing*. The designer’s moves are a combination of all these experimentations at once.

Exploratory experiments are described to be experiments where action is undertaken “*to see what follows, without accompanying predictions or expectations*” (Schon, 1984, p. 145). It is described as a playful activity, to “*get a feel for things*”. It succeeds when “*it leads to the discovery of something there*” (Schon, 1984, p. 145).

Move-testing experiments are experiments where action is undertaken to produce an intended change. If a move results in an intended change, it is described to be *affirmed*, otherwise, it is described to be *negated* (Schon, 1984, p. 146).

Hypothesis testing experiments are experiments which discriminate between competing hypotheses. These involve actions which aim to confirm hypotheses by performing actions with predicted consequences. Hypotheses with predicted consequences are described to be *confirmed*, whereas others are *disconfirmed* (Schon, 1984, p. 146).

BACKGROUND

Experimentation is essential in both descriptions of design activity. It enables the designer to extract information from a problem to make informed steps to arrive at a solution.

DESIGN TOOLS

The Use of Representation

The ability to represent information externally through symbols and drawings provide designers with the means to solve problems by a) extending the limitations of their memory and attention and b) enabling designers to infer the results of design experiments.

When designing, the designer has more information than they are able to attend to at one moment. The use of external representations enables the designer to examine the information at will and at their own pace (Schon, 1984, p. 157 - 158). These representations take the form of drawings and diagrams, which act as a store of information for a given problem (Simon, 1995). Information is added selectively to the drawing as relevant information is retrieved or generated. It enables designers to make sense of the situation by bringing the relevant details to attention while omitting the irrelevant.

Representation is also essential in enabling rigorous design experimentation by making the information required to solve geometric problems explicitly recognisable (Larkin & Simon, 1987). Simon (1995, p. 249) describes the ability of representations to create objects and their relationships diagrammatically to make them inference engines which “*effortlessly ‘calculates’*” the consequences of recorded actions. Additionally, simplified representations enable designers to alleviate the constraints which prevent or inhibit experimentation. Schon (1984, p. 157-158) expresses this in his concept of a *Virtual World*, — “*a constructed representation of the real world of practice*”. In the context of architectural design, he describes that

“moves that would be costly in the real world can be tried at little or no risk”.

SPACESHEETS: DESIGN EXPERIMENTATION IN LATENT SPACE

However, since Virtual Worlds are simplifications of the real situation, there are limitations in the transferability of the results of experiments. Schon (1984, p. 159) addresses this by noting that “*the validity of transfer depends on the reliability with which the drawn world represents the built one*”, and that the designer “*must remember the factors that have been eliminated*” when interpreting the results of their experiments.

Representations enable designers to manage the complexity of a situation and experiment rigorously. A goal of design software is to support designers by enabling methods of constructing and manipulating representations digitally.

Trade-offs in Design Tools

Design tools in this section refer to a subset of the tools used in a design process. Specifically, software which enables users to represent and perform operations on geometry. There is a myriad of design tools available, each designed for different goals, resulting in different trade-offs.

Distinctions across two dimensions have been made for the purposes of discussion: *parametric* vs. *direct manipulation*; *high-level* vs. *low-level* operations.

Parametric design tools enable users to define a model — a collection of objects and relationships which make up a design. Examples of parametric tools include *SolidWorks*, *Rhino with Grasshopper* and *Autodesk Revit*. Designers can modify parameters within the model to arrive at objects which are consistent with the user-defined relationships. This enables designers to explore a range of design variations easily within meaningful constraints.

The user interfaces for parametric tools generally take two forms: text-based, within *integrated development environments* (IDE) in software such as *Processing* and *OpenFrameworks*, or within a *graphical user interface* (GUI) such as in *Grasshopper*.

Direct manipulation enables users to create and modify representations and see its implications immediately. Examples of direct manipulation tools include *Adobe Photoshop* and *GIMP*. Designers are able to create and modify representations using operations based on metaphors such as brush, pen and eraser tools. The effects of these operations are immediately visible and made to be “*rapid, incremental, reversible*” (Shneiderman, 2017, p. 200). The benefit of this immediacy makes it easy for user

BACKGROUND

“see whether their actions are furthering their goals” and to *“simply change the direction of their activity”* if their actions are counterproductive (Shneiderman, 2017, p. 201).

Between parametric and direct manipulation tools, a trade-off emerges between the *ease of design experimentation* and the *ease of building a model*. In parametric tools, it is much easier to explore a range of meaningful design variations which fit within the designer’s encoded interrelations between geometry.

This comes at an upfront cost of having to build a model. It involves skills which are complex and idiosyncratic, requiring the designer to operate in foreign ways. Woodbury states that building a model *“...requires a formal notation and introduces additional concepts that have not previously been considered as part of ‘design thinking’”* (Woodbury, 2010, p. 24).

It is much easier to build a design through direct manipulation. However, this mode of working is critiqued to be a mismatch to the flexibility required in design processes, described to *“limit exploration and effectively restrict design”* (Woodbury, 2010). Terry and Mynatt (2002, p. 38), attribute this to the interface needing to be in *“one, and only one, state at any particular time”*, referring to the type of interface as *The Single State Document Model*. They reason that these interfaces impose a *“serial, linear progression through a task that is at odds with the ‘messy’, highly iterative process”* (Terry and Mynatt, 2002, p. 38).

High-level and *low-level* operations refer to the level of abstraction of an operation afforded to the end user by a design tool. A high-level operation can be achieved by several low-level operations. For example, a high-level operation may be to make the image ‘warmer’ in colour. The effect of this operation can be achieved by a sequence of lower-level operations which increases the red value of each pixel.

The trade-off in high-level and low-level operations exists between *expressiveness* and *efficiency*. High-level operations are more efficient but are inflexible as they rely on predetermined low-level settings. Low-level operations are more expressive, but inefficient. In the previous example, a preset ‘warming’ effect would be much faster than increasing each pixel’s red value. However, this is limited in its flexibility by its predetermined settings — of which pixels should be red, and by how much, etc.

SPACESHEETS: DESIGN EXPERIMENTATION IN LATENT SPACE

Hebron (2017) identifies this trade-off in the differences between consumer-level and professional design tools. He describes that consumer-level tools “*simplify design processes by forcing users into one of a handful of preordained templates*” and professional design tools “*provide an overwhelming number of low-level features that come with steep learning curves and often do not coincide with the user’s way of thinking*”.

These trade-offs are in part due to design software being built on the constraints of deterministic programming. They are “*designed to operate on logical expressions that can be evaluated without the knowledge of any outside factor beyond those expressly provided to them*” (Hebron, 2016, p. 2).

Machine learning algorithms enable “*mechanisms for imparting experiential knowledge upon computing systems*” (Hebron, 2016, p. 2), they show promise in enabling computers to support designers in more natural ways.

Design Tools and Machine Learning

Machine learning opens the possibility to address the limitations of current design tools, as well as enable new avenues of supporting the design process.

This section summarises the articles *CreativeAI: On the Democratisation & Escalation of Creativity* by Pieters & Winiger (2016) and *Rethinking Design Tools in the Age of Machine Learning* by Hebron (2017) with respect to the opportunities provided by machine learning.

CreativeAI: On the Democratisation & Escalation of Creativity (Pieters & Winiger, 2016) investigates the history of technology and creativity, and posits a “*vision for a future where CreativeAI helps us raise the human potential*”.

Pieters and Winiger make two distinctions in which technology helps us to create: *Assisted Creation* and *Generative Creation*.

Assisted Creation is described as the use of technology to “*assist humans to perform ‘creative’ tasks*”. These take the form of tools such as *Autotune*, which performs pitch-correction for music and *Autocorrect* which corrects spelling errors for writing.

They quote Shneiderman (2000, p. 115), describing the technology as allowing

BACKGROUND

more people *“to be more creative more of the time”*. This is achieved by *“lowering the bar of mastery”*, enabling users to *“shift their attention to higher-level issues, perform complex creative tasks more reliably and experiment quickly”*.

They note, however that these tools can be limiting to creativity, in that the *“negotiation for control is blunt and interactions not fine-grained”*. To this, they articulate a vision for machine learning to create *“systems that negotiate the creative process in fine-grained conversations, augment creative capabilities and accelerate the skill acquisition time, from novice to expert”*.

Generative Creation is described as a *“shift from object to process”*, in which we use models — *“computational representation[s] of reality”* — to generate artifacts. They quote Hansmeyer (2012) who describes the process as *“thinking about designing not the object but a process to generate objects”*.

They note that machine learning extends the capability of creating models for problems, *“allowing us to model complexity with greater resolution and apply modelling techniques to a wider range of creative problems”*. These sophisticated models can be then be explored and visualised, enabling us to *“easily retrieve information, explore data and ask questions about relationships, logic and meaning”*.

Rethinking Design Tools in the Age of Machine Learning by Hebron (2017) identifies the implications of machine learning on the capability of design tools.

Hebron describes several possibilities in the sections titled: *Emergent Feature Sets*, *Design Through Exploration*, *Design by Description* and *Process Organisation and Conversational Interfaces*.

Emergent Feature Sets explains the potential for machine learning algorithms to learn sophisticated high-level operations from user interactions. The enables the tool to be better suited to the *“diversity of designers and their varied ways of digesting information, making decisions and interacting with software”*.

Design Through Exploration explains how machine learning algorithms are able to organise designs spatially, enabling users to explore a search-space of design artifacts. Designers would be able to explore this space directly to realise design intent instead of using sequences of low-level operations. This reduces the cost

SPACESHEETS: DESIGN EXPERIMENTATION IN LATENT SPACE

of design exploration.

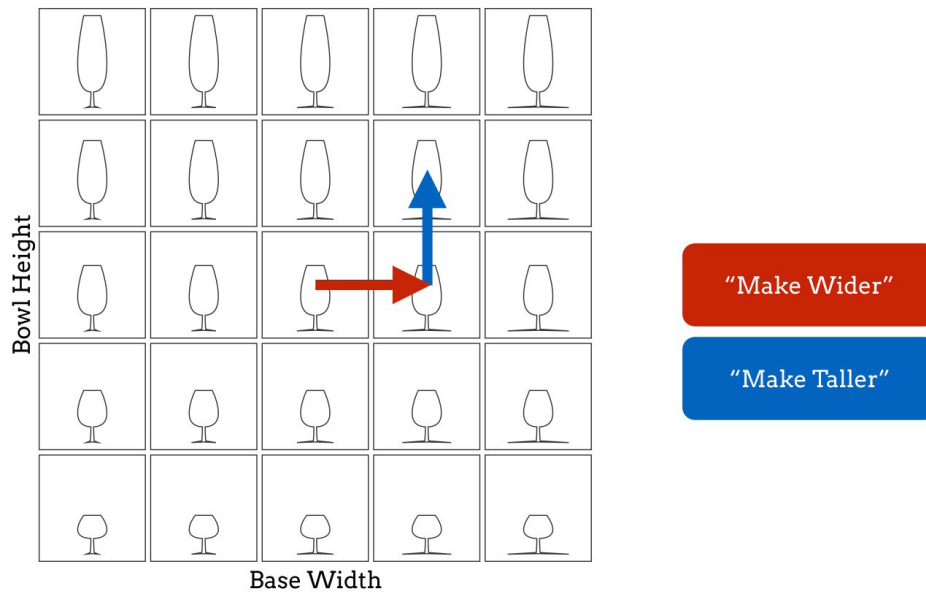


Figure 1: A two-dimensional space of wine glasses.

Operations are performed to transform its form.

Source: Drawing by Patrick Hebron (2017). *Rethinking Design Tools in the Age of Machine Learning*. Retrieved from URL: <https://medium.com/artists-and-machine-intelligence/rethinking-design-tools-in-the-age-of-machine-learning-369f3f07ab6c>

Design By Description explains how machine learning is able to compute the relationships between words. Hebron proposes that this can be used as a mechanism to explore the search spaces described in the *Design Through Exploration* section.

This mechanism enables designers to explore and compose ideas through direct manipulation of the concept space in which these ideas reside, giving the example of subtracting the cubist aesthetic of Braque from Picasso in an attempt to find “an aesthetic that was like Picasso’s but not from the high of his Cubist period”.

Process Organisation and Conversational Interfaces describes how design tools can be used to break down design processes into a series of “simple exercises or decision points that each address a single facet of a much larger and more complex task”. This is achieved

BACKGROUND

through an “*algorithm called a Learning Decision Tree to determine the sequence of questions that will lead to the correct answer in the smallest number of steps possible*”.

This organisation would also enable designers to “*inspect each stage in his or her thinking and easily return to earlier iterations, branching off in a new direction while still preserving each other version of the design*”.

The two articles summarised express the potential for machine learning to create tools which are better suited to support the design process. They describe a diverse set opportunities in which the limitations of existing design tools can be addressed to enable designers to be more creative.

SAMPLING LATENT SPACES

Generative neural networks are a class of machine learning algorithms which learn to construct a generalised model from a set of training data. These models can be sampled to generate new data samples which resemble the data it has been trained on. This is achieved by training the neural networks on a smaller set of parameters than that within the original data. By doing so, the model learns more compact representations of the training data. These representations are called *latent variables*. Latent variables are sampled from *latent space* can be decoded back into observable data samples.

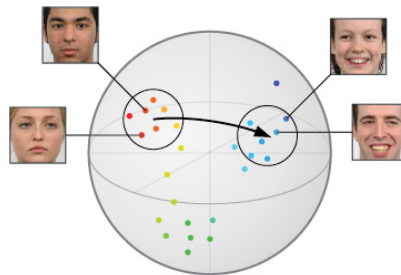


Figure 2: Diagram of a latent space of faces. Points within the latent space can be decoded into observable images of faces.

Source: Image by Tom White (2017). *Sampling Generative Networks*.

Retrieved from URL: <http://arxiv.org/abs/1609.04468>

SPACESHEETS: DESIGN EXPERIMENTATION IN LATENT SPACE

As latent variables are more compact, they learn a more abstract, higher-level model of the training data. This makes it possible to “*generalise beyond the training examples already seen*” to generate realistic looking data (Carter and Nielsen, 2017).

The model captures heuristics and infers principles implicit within data. This enables sophisticated transformations to be expressed in latent space. Carter and Nielsen (2017), demonstrate this by comparing the differences of a naive bold operation, — in which pixels are added around the edges of the glyph — and an expert bold operation — which is less trivial, involving modifications to its structure. This operation is then expressed in latent space, resulting in a glyph which resembles an expertly bolded glyph.

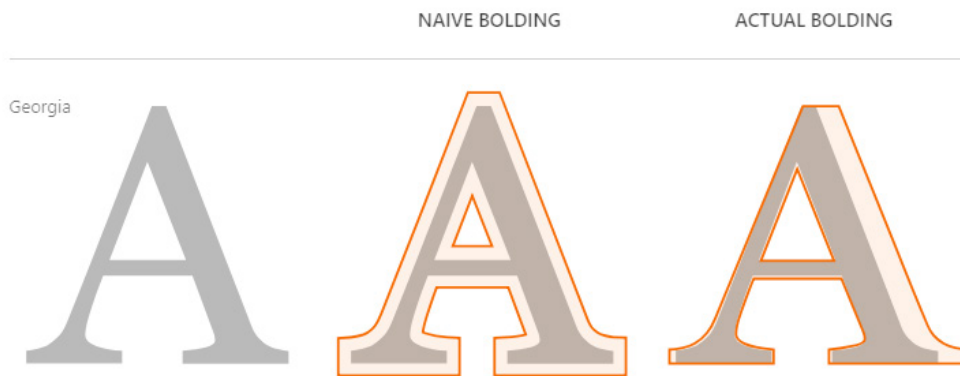


Figure 3: Naive bold comparison with actual bolding.

Source: Drawing by Shan Carter & Michael Nielsen (2017). *Using Artificial Intelligence to Augment Human Intelligence*. Retrieved from URL: <https://distill.pub/2017/aia/>

Operations can be expressed between latent variables through vector space arithmetic. These operations result in high-level semantic differences when its results are decoded back into observable space. A variety of techniques have been established to sample and visualise latent spaces meaningfully. These techniques include *analogy*, *interpolation* and *attribute vectors* (White, 2016).

Analogies are formally noted as “ $A:B :: C:?$ ”. This asks the question “*What is the result of applying the transformation $A:B$ to C ?*” (Reed et al., 2015). In latent spaces, this can be solved by using the arithmetic: “ $? = (B - A) + C$ ” to infer the results of pairwise differences. Analogy has been used in linguistic models to the effect

BACKGROUND

of “*King - Man + Woman \approx Queen*” (Mikolov et al., 2013) as well as to create visual analogies (Reed et al., 2015).

Interpolation is the process of sampling intervals between two discrete points. It is used in the research of generative models, to demonstrate that a model has not memorised training examples, as well as in creative applications to provide smooth transitions between two images (White, 2016).

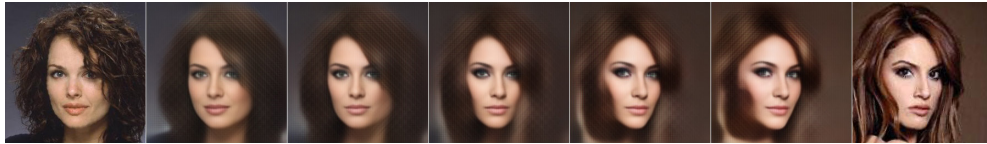


Figure 4: An interpolation between two faces.

Source: Image by Tom White (2017). *Sampling Generative Networks*.

Retrieved from URL: <http://arxiv.org/abs/1609.04468>

Specific attributes can be isolated from latent variables by subtraction. The result of these subtractions are termed *attribute vectors* and can be applied to latent variables to transform it towards being more like the attribute. For example, Larsen et al. (2015) construct a “*smile vector*” in a latent space of faces by calculating the mean vector for images of smiling faces and subtracting the mean vector for images of faces without smiles. This vector can then be added to or subtracted from latent variables to make them more or less ‘smiley’.



Figure 5: Applying a “smile vector” to a face.

Source: Image by Tom White (2017). *Sampling Generative Networks*.

Retrieved from URL: <http://arxiv.org/abs/1609.04468>

Although attribute vectors are effective in capturing semantic concepts, they are known to suffer from ‘correlated attributes’ (Larsen et al., 2015), which reveal strong correlations between several attributes. For example, White (2016) discovered that the application of a smile vector resulted in adding more feminine

SPACESHEETS: DESIGN EXPERIMENTATION IN LATENT SPACE

attributes to the sample. This can be caused by either sampling bias in calculating the mean vectors to compute the attribute vectors (White, 2016), as well as bias in the dataset (Larsen et al., 2015).

HYPOTHESIS

HYPOTHESIS

Experimentation is the foundation of design — to search a space in *Rational Problem Solving* (Newell & Simon, 1972), and to transform the current situation to a more desirable situation in *Reflective Practice* (Schon, 1984). Between these two paradigms, there are several approaches to experimentation: *Hill Climbing*, *Means-end Analysis*, *Exploratory Experimentation*, *Hypothesis Testing*, *Move Testing*.

Experimentation is enabled through representations in the form of drawings and diagrams. These representations are simplified models of the situation, enabling designers to comprehend the complexity of design situations and to alleviate constraints which limit experimentation.

Computational design tools enable users to construct and manipulate representations digitally. These tools often impose a high cost to design experimentation. *Parametric* tools enable users to experiment flexibly within meaningful design constraints at the expense of the upfront effort and skill required to construct a model. Conversely, *direct manipulation* tools enable users to construct models easily at the cost of inefficient design experimentation.

This trade-off between design the ease of design experimentation and the ease of building a model is in part due to the constraints of the low-level operations afforded by deterministic programming. Users are either limited to expressing design intent through extended sequences of low-level operations, or inflexible predetermined high-level operations.

Machine learning algorithms provide “*mechanisms for imparting experiential knowledge upon computing systems*” (Hebron, 2016, p. 2), and show promise in creating tools which support designers in more natural ways.

Generative neural networks are able to construct generalised models of images which capture principles implicit within the images. The latent spaces of these models can be sampled to create novel images and to perform semantic operations.

The ability to sample a model of design artifacts provides the opportunity for designers to express design intent more meaningfully through the principles

SPACESHEETS: DESIGN EXPERIMENTATION IN LATENT SPACE

inferred by the model. A system which affords this ability is promising as a tool for design experimentation.

The tool presents a novel top-down where users are provided with a predetermined model to experiment within, as opposed to constructing a model from the bottom-up to experiment with.

A new mode of working poses a new set of challenges. New user intuition is required to use the tool effectively. Methods of sampling latent spaces, — *interpolation* and *attribute vectors* — are not part of the conventional toolkit for design.

The opportunity for this tool and the challenges that accompany it form the context for the design problem.

DESIGN PROPOSITION

DESIGN PROPOSITION

DESIGN PRINCIPLES

This section documents a collection design principles to be used as heuristics for making decisions during the design process. These principles are selectively curated from *Design Principles for Tools to Support Creative Thinking* by Resnick et al., (2005) and supported by ideas proposed in *Recognizing Creative Needs in User Interface Design* by Terry & Mynatt (2002) to serve the primary design goal of supporting experimentation and exploration.

Although useful, principles are general and detached from the specifics of the project. For this reason, they will not be used prescriptively, but as guidelines to focus design efforts.

These principles can be summarised by the three user interface requirements proposed in *Design Principles for Tools to Support Creative Thinking* (Resnick et al., 2005) to support exploration (paraphrased):

1. *It must be very easy to try things out and then backtrack when unsuccessful.*
2. *Tools should be ‘self-revealing’ in what they can achieve.*
3. *Make it very fast to sketch out different alternatives.*

The first requirement: “*It must be very easy to try things out, and then backtrack when unsuccessful*” is fundamental in allowing users to explore and experiment. Terry and Mynatt (2002), provides further insight into this requirement through the perspective of Schon’s theory of *reflection-in-action* (1984). They identify three activities in the process of reflection-in-action and describe how they can be supported by user interfaces. These activities are *Near-Term Experimentation*, *Variations*, and *Evaluation*.

Near-Term Experimentation is used to describe actions which intend to “*discover and instantiate the next move*” (Terry & Mynatt, 2002, p. 39). In a user interface, users would make hypotheses about the next action to be made, and test their hypothesis

SPACESHEETS: DESIGN EXPERIMENTATION IN LATENT SPACE

by “*invoking a command and adjusting its settings to achieve the imagined effect*”. The users would then “*either accept the command, tweak the parameters more, or undo it and try another tact*” (Terry & Mynatt, 2002, p. 40).

Near-Term experimentation is supported by the capability to undo actions and tight feedback loops between action and effect through direct manipulation (e.g. slider widgets) (Terry & Mynatt, 2002, p. 41).

Variations are created by the designer to explore alternatives deeply. It enables them “*to better understand the problem, its boundaries, and potential solutions*” (Terry & Mynatt, 2002, p. 40). An example of this is where designers make “*multiple variations of a specific component by creating them side-by-side on a large canvas...and iterate on promising versions to arrive at an acceptable solution*” (Terry & Mynatt, 2002, p. 40).

The user can be supported in the process of generating variations by “*mechanisms that let users duplicate their data*”, providing space in the document to hold multiple versions, and providing the ability to undo (Terry & Mynatt, 2002, p. 41).

Users need to *evaluate* their progress as they work on a task. This happens after near-term experiments, as well as after generating variations (Terry & Mynatt, 2002). Terry and Mynatt describe this as “*the moment in which the individual reassesses the problem and their understanding of it, before making the next move*” (Terry & Mynatt, 2002, p. 40).

The evaluation process can be supported by providing “*multiple perspectives and views, such as different levels of zoom, and through alternative representations*” (Terry & Mynatt, 2002, p. 42).

The second requirement proposed by Resnick et al. (2005) to support exploration is that the *tools should be ‘self-revealing’*, in what users are able to achieve with it. This is important when users are learning the tools. They add on the requirement for tools to be “*facile and unencumbering*”, to enable expert users to “*try out different alternatives very quickly*”.

The third requirement to support exploration is to “*make it very fast to ‘sketch’ out different alternatives at the early stages of design*”. This enables users to try out different ideas before settling on an implementation.

DESIGN PROPOSITION

SPREADSHEET AS A DESIGN TOOL

Spreadsheet interfaces enable users to enter data and formulae into a grid of cells. These formulae enable users to define relationships between cell values. As data changes, the spreadsheet is updated and all of its values are recalculated.

Work Item	Vendor	Labor	Equipment	Materials	Subcontr.	Subtotal	Markup %	Markup	Total
Permits/Fees	City of Los Angeles				\$1,500.00	\$1,500.00		\$0.00	\$1,500.00
Excavation		\$6,000.00	\$8,000.00	\$500.00		\$14,500.00	15.00%	\$2,175.00	\$16,675.00
Utilities		\$3,500.00	\$2,500.00	\$2,750.00	\$1,000.00	\$9,750.00	15.00%	\$1,462.50	\$11,212.50
Water Well						\$0.00		\$0.00	\$0.00
Septic Tank						\$0.00		\$0.00	\$0.00
Foundation	Connie's Concrete				\$3,500.00	\$3,500.00	5.00%	\$175.00	\$3,675.00
Concrete Flatwork	Connie's Concrete				\$1,900.00	\$1,900.00	5.00%	\$95.00	\$1,995.00
Framing		\$3,500.00	\$1,500.00	\$9,000.00		\$14,000.00	15.00%	\$2,100.00	\$16,100.00
Roofing	Robert's Roofing				\$3,500.00	\$3,500.00	5.00%	\$175.00	\$3,675.00
Windows/Ext. Doors	Wally's Windows				\$8,000.00	\$8,000.00	5.00%	\$400.00	\$8,400.00
Garage Door	Gary's Garage Doors				\$2,250.00	\$2,250.00	5.00%	\$112.50	\$2,362.50
Siding						\$0.00		\$0.00	\$0.00
Electrical	Ernie's Electric				\$18,500.00	\$18,500.00	5.00%	\$925.00	\$19,425.00
Plumbing	Mac's Mechanical				\$16,500.00	\$16,500.00	5.00%	\$825.00	\$17,325.00
HVAC	Mac's Mechanical				\$23,000.00	\$23,000.00	5.00%	\$1,150.00	\$24,150.00
Insulation		\$3,500.00		\$1,000.00		\$4,500.00		\$0.00	\$4,500.00
Masonry	Mason's Masonry				\$14,500.00	\$14,500.00	5.00%	\$725.00	\$15,225.00
Drywall	Doug's Drywall				\$12,500.00	\$12,500.00	5.00%	\$625.00	\$13,125.00
Interior Trim	Doug's Drywall				\$9,000.00	\$9,000.00	5.00%	\$450.00	\$9,450.00
Painting	Paul's Painting				\$13,500.00	\$13,500.00	5.00%	\$675.00	\$14,175.00
Floor Coverings	Carl's Carpets				\$16,500.00	\$16,500.00	5.00%	\$825.00	\$17,325.00
Cabinets	Ken's Cabinets				\$22,500.00	\$22,500.00	5.00%	\$1,125.00	\$23,625.00

Figure 6: Spreadsheet software being used to create a cost estimation.

Source: Wikipedia. A cost estimate spreadsheet prepared with computer spreadsheet software (2011). Retrieved from URL: https://commons.wikimedia.org/wiki/File:Estimating_Spreadsheet.png

A spreadsheet interface which enables users to operate on latent variables shows promise as a flexible, low-level tool for design experimentation. The distinction here is that instead of numerical data, cells would hold latent variables and the operations between these cells would be vector arithmetic. Cells which hold latent variables will be displayed as their respective decoded image.

Spreadsheets may seem like an unlikely design tool. However, the ability to express relationships between cells make it functionally suited to express operations in latent space.

Additionally, it satisfies the three user requirements for software to support design experimentation — *Near-Term Experimentation*, *Variations*, and *Evaluation* (Terry & Mynatt, 2002) — as identified in the design principles section.

SPACESHEETS: DESIGN EXPERIMENTATION IN LATENT SPACE

Near-term experimentation is supported by the automatic updating feature of the spreadsheet. Users are able to set up scenarios of logic and calculate the results to ‘*what if*’ questions instantly by modifying the cell values. This establishes a tight feedback loop between the user’s actions and its implications. When coupled with the ability to undo actions, it enables users to discover and instantiate moves, and backtrack if the results are unsatisfactory.

The generation of *variations* is supported by enabling users to duplicate instances of data onto other cells within the document. These copies can then be modified independently from the original data.

Evaluations are supported by enabling users flexibility in how they choose to organise data in the document. Users can set up custom templates in a layout which best supports their preferences and the problem to be solved.

In addition to their promise in supporting design experimentation, spreadsheet software is well-established within office productivity suites. Users with an understanding of how conventional spreadsheets function are able to transfer their understanding to the use of the design tool.

DESIGN PROCESS

DESIGN PROCESS

DEVELOPMENT APPROACH

In addition to the design principles, the development approach will determine what will be created. Here are the general principles I will follow.

Build to Generate Knowledge

Most importantly, the design will be created to generate knowledge of the suitability of latent spaces as a design tool. In the case of this research, knowledge is generated through user testing. Development tasks which support testing design hypotheses will be prioritised over other tasks.

Build Idealistically

Interfaces for latent spaces and machine learning enabled design tools are a relatively new field. As such, there are no established best practices for technical implementation. There will be more and less optimal technical approaches. However, discovering these technical details is not the priority of the research. They only matter so far as the design can be implemented and function as intended.

Additionally, the quality of latent spaces may be prone to generate samples with artifacts or be of a low resolution. Instead of being paralysed by the limitations of the current state of technology, an idealistic viewpoint will be taken. This viewpoint acknowledges the limitations of the latent space as something that will be improved over time with further research in the field.

SPACESHEETS: DESIGN EXPERIMENTATION IN LATENT SPACE

ADDITIONAL CONSIDERATIONS

In addition to the design principles, several additional considerations will be made to inform the implementation of the tool.

Accessibility

Latent spaces are highly inaccessible at the moment. This is due to the need for particular software environments. Specific software environments have to be preconfigured to load and generate samples from latent space. Furthermore, specialised knowledge of the codebase and programming is required.

Alleviating the need for programming knowledge is part of the design problem, and the need for specific software environments would be a nice-to-solve.

A General Purpose Tool

Generative models have been used to model a diverse range of images — logos (Sage et al., 2017), materials (Zsolnai-Fehér et al., 2018), etc. Since these models are abstracted away from the specifics of the output, it is possible to design the tool to operate with a variety of latent spaces with minimal reconfiguration.

INITIAL SPECIFICATIONS

The concept of a spreadsheet interface as a tool to explore latent space will be used as the starting point for the process. At its minimum, the design will require:

1. A latent space to explore.
2. An interface for users to input latent variables from the space into the spreadsheet. This interface element is referred to as the *Data Picker*.
3. A spreadsheet interface adapted to support meaningful latent space exploration.

DESIGN PROCESS

The Latent Space

With regards to the software, the latent space takes the form of the pre-trained weights of a generative neural network model. These weights are then used to decode latent variables into images.

The latent space chosen for the project is Eric Bernhardsson's *DeepFonts*, a generative model for fonts (Bernhardsson, 2015/2018). It is chosen as fonts are a suitable design output to experiment with: there are established typographic properties — serif, bold, condensed.



Figure 7: An interpolation between two glyphs in the DeepFonts model.

Additionally, the file size of its weights is relatively small compared to other alternatives. This enables it to be downloaded quicker, making for a better online user experience.

Data Picker

Latent variables are high-dimensional vectors — a list of cryptic numbers (40 numbers long in the case of the DeepFonts model). Unlike numbers in a spreadsheet, latent variables cannot be expressed meaningfully or efficiently through textual input. They have to be decoded into images to make sense to the user.

For this reason, a more intuitive interface is necessary to support meaningful user input. The *Data Picker* presents a predetermined set of latent variables as a grid of decoded images. Clicking on an image populates cells in the spreadsheet with its corresponding latent variable value.

The latent variables within the Data Picker are predetermined points of references in the latent space from which the user can operate with. Several considerations are made when choosing these variables.

SPACESHEETS: DESIGN EXPERIMENTATION IN LATENT SPACE

Firstly, the variables are the design primitives of the tool, and will greatly determine what the user is able to achieve with the tool.

Secondly, there is a risk of providing points in latent space which are *“highly unlikely given the prior of the model”* (White, 2016, p. 2), which result in images unlike those in the training set.

To account for these factors, a diverse set of reconstructions — encoded images of fonts from which the model is constructed from — is selected as the variables in the Data Picker.

For added flexibility, the Data Picker can be zoomed in upon to subdivide the grid of images. Subdivisions will reveal intermediates interpolated between points within the grid. This is inspired from the design principle to *“make it very fast to sketch out different alternatives”*.

Spreadsheet

The spreadsheet enables users to express relationships between latent variables. Several modifications to conventional spreadsheet features are specified for the design.

Most obviously, the spreadsheet enables users to operate between references in latent space, (expressed through the Data Picker element) through vector arithmetic. Cells with latent variables are displayed as their decoded images. Numeric operations and alphanumeric values are still supported.

Operations on latent variables will be interfaced through the ‘operation-parentheses syntax’ e.g. *“SUM(A, B)”*, instead of *“A + B”*. This is to encourage users to break processes down into decomposable steps.

Operations to perform linear interpolation and calculate the Euclidean distance between two points are implemented to support design experimentation and user understanding.

DESIGN PROCESS

A slider cell is specified to enable users to build workflows with tight action-feedback loops which support reflection-in-action. These cells are interpreted as numbers but are controlled by a slider element.

A random font cell is designed to negate the limitations of the predefined selection of latent variables in the Data Picker. These cells conjure a random font reconstruction from the generative model through the *RANDFONT* operation. A random seed is used as a parameter in the operation, to enable users to record fonts of interest.

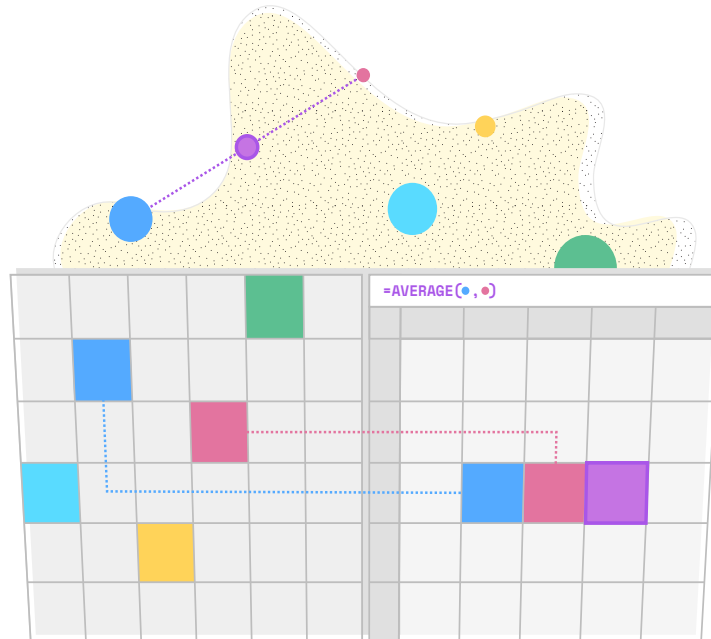


Figure 8: Diagram of the SpaceSheet. The Data Picker (front left) is used to populate cells in the spreadsheet (front right) with variables from latent space (back). The spreadsheet can be used to define operations to arrive at new points in latent space.

SPACESHEETS: DESIGN EXPERIMENTATION IN LATENT SPACE

TECHNICAL SPECIFICATIONS

The software is built as a web application. This enables the tool to be accessible through any computer with an internet browser.

The decoding of the latent variables will be executed in-browser, using *deeplearn.js* (now *Tensorflow.js* ('TensorFlow.js', n.d.)). Outsourcing the computation to a web server had been considered, as it would enable the use of commonly used statistical and machine learning libraries. However, experiments in *deeplearn.js* revealed it to be sufficient to achieve everything required by the design specifications performantly.

The Latent Space

The DeepFonts model is loaded from an external source as the application initialises. The model parameters, — vector length, image size, encoding and decoding functions — are specified in a separate file to be read by the application before rendering. This is to enable the application to be used to explore other latent spaces with minimal reconfiguration.

The Interface

The user interface is built using the *React.js* (react, 2013/2018) framework. The Data Picker and spreadsheet components are built using *HTML Canvas* and *HandsOnTable.js* ('Handsontable - JavaScript Spreadsheet Component For Web Apps', n.d.) respectively.

The Data Picker is built to render a grid of decoded latent variables from an external file. This file is created in a separate process by running the *smartgrid* (smartgrid, 2017/2017) script on a set of curated images. *smartgrid* is a script which clusters a set of images and assigns them into a gridded layout.

HandsOnTable.js is a customisable spreadsheet component. It contains the core features of a spreadsheet — cells, selection, autofill, etc. — and can be customised with a great deal of flexibility. However, several modifications will have to be made for the design. These include the ability to perform vector arithmetic between

DESIGN PROCESS

latent variables, custom operations and a custom cell to display latent variables as their decoded images.

PROOF OF CONCEPT: COLOUR SPREADSHEET

To test the assumptions of the design specifications, a rough prototype is devised to explore an RGB colour space. This analogy is functionally equivalent to latent space and serves to test the design efficacy of the spreadsheet interface without the burden of implementing the generative model.

It is, of course, more effective to explore the colour space with one of the many colour pickers available. However, the goal is not to enable effective colour space exploration, but to get a first impression of the nature of the spreadsheet interface as a design tool.

The proof of concept is also an opportunity to create a first draft of the codebase. It will build an understanding of the frameworks used and the challenges in integrating them together. This will be helpful in making better development decisions when building the *SpaceSheet*.

The extent of the analogy is as such: instead of high-dimensional latent variables, the variables are three-dimensional RGB colour values, and instead of decoding the latent variable into images, the cell is rendered as its corresponding RGB colour.

The spreadsheet component is created in this implementation, however, the Data Picker will be replaced with a standard colour picker (Sandberg, 2015/2018) for the purpose of rapid prototyping.

SPACESHEETS: DESIGN EXPERIMENTATION IN LATENT SPACE

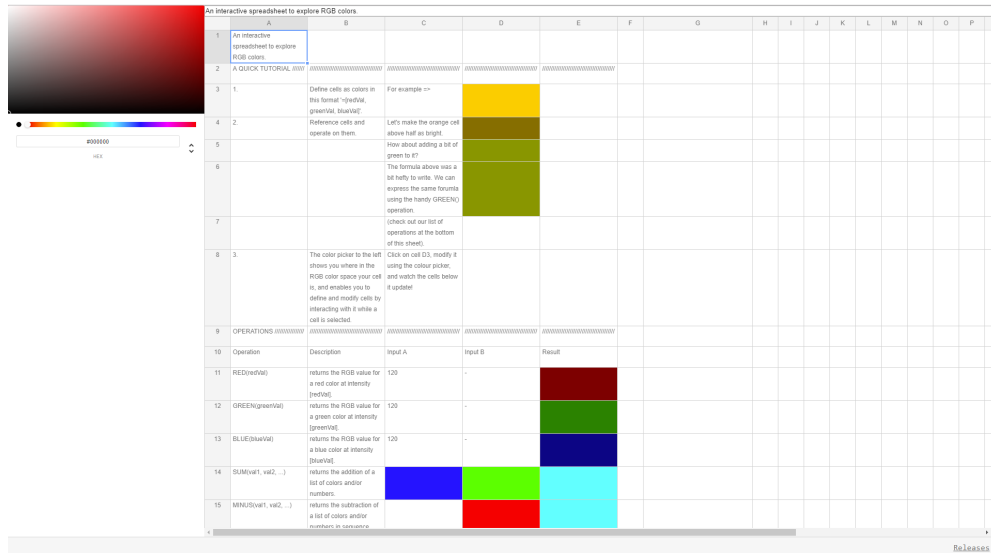


Figure 9: Screenshot of the Colour Spreadsheet.

This exercise revealed that the spreadsheet interface is a flexible tool for experimentation. This is mainly due to the combined effect of two features: the ability for users to define sequences of operations and its ability to update automatically.

The slider cell also proved to be a useful control for parameters. It enables users to have fine control over the effect of a parameter, as its results updated instantly. For example, a slider was effective in controlling the degree of interpolation between two cells.





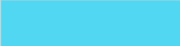











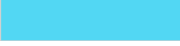



	A	B	C	D	E
1	From	To	Amount	Slider Value	Result
2				0	
3				0.25	
4				0.5	
5				0.75	
6				1	

Figure 10: A series of interpolations between two colours with the slider element.

DESIGN PROCESS

A *RANDCOLOR* operation was created as an analog to the *RANDFONT* operation in the design. This creates a cell with a random colour. This was hard to control. Using a random seed could be a solution to this.

A summary of insights gathered from the proof of concept:

- The spreadsheet interface was effective in enabling users to experiment flexibly.
- The slider cell was useful in providing instant feedback and fine control.
- The random color cells were hard to control.

SPACESHEETS: DESIGN EXPERIMENTATION IN LATENT SPACE

VERSION ONE

This version includes the developments which precede the first round of user testing. It is built from scratch, with the experience gathered from the colour proof-of-concept.

For the most part, this version of the application resembles the initial design specifications. The differences are in a few modifications that are implemented in the process of building and testing the application. Most notably these include the way in which operations are defined by the user and cell highlighting.

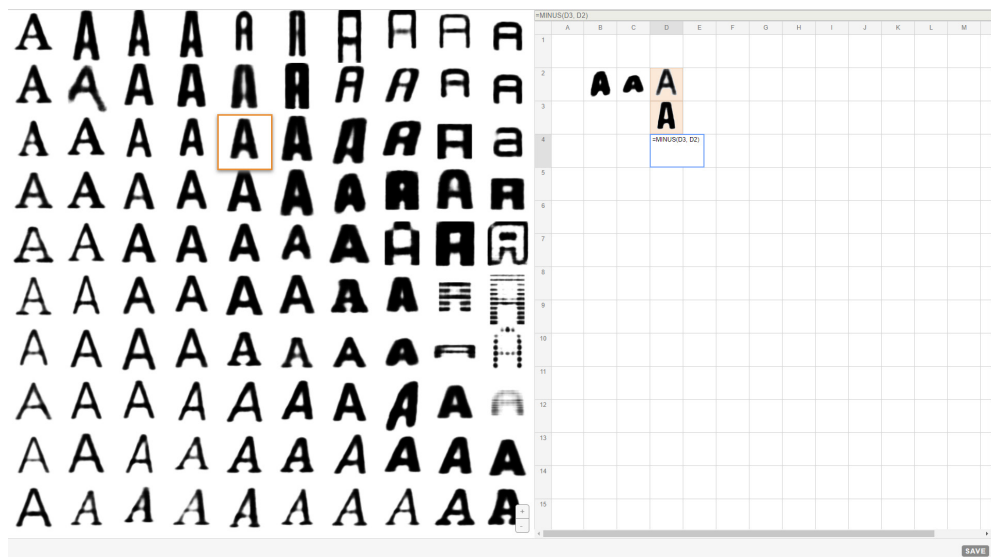


Figure 11: Screenshot of Version One.

DESIGN PROCESS

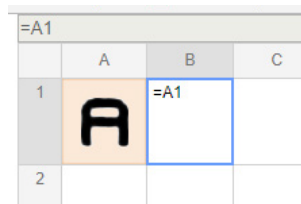
Operation Aliasing and Defaults

Several operations are aliased. For example, the *ADD* operator is equivalent to *SUM*. Both words have the same intent and are equally valid. Users are able to express either operation without any repercussion. This is inspired by the notion of “*do as I am intending, not as I am doing*” (Norman, 1986, p. 51).

In a similar vein, if operations are expressed with no parameters, they are populated with probable default values. For example, a *=SLIDER()* operation with no parameters is prepopulated to start at 0 and end at 1. This range is chosen as the slider element is most likely used to interpolate between two fonts.

Cell Highlighting

When using the spreadsheet, I found myself frequently locating cells by its reference e.g. ‘A2’. This process was tedious and often detracted from the process of design experimentation. To deal with this, cell highlighting as it exists in conventional spreadsheet software is implemented. As a cell is being edited, the spreadsheet highlights all the cells which the cell value references. Highlights are updated as the user modifies the cell value, making it obvious if the user has entered an unintended reference for a cell.



	A	B	C
1	A	=A1	
2			

Figure 12: Cell references in cells being edited are highlighted.

SPACESHEETS: DESIGN EXPERIMENTATION IN LATENT SPACE

RANDFONT Cell

Developing from the previous iteration, the *RANDFONT* cell is redesigned to use a random seed. This enables it to be controlled more predictably and also enables the user to interact with it via a slider element.

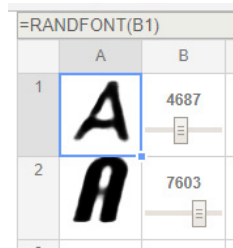


Figure 13: Controlling the RANDFONT cell with a slider element.

A button displays when the user mouses over the cell. When clicked, the cell re-randomises by generating a new random seed.



Figure 14: The RANDFONT cell can be randomised by clicking a button.

DESIGN PROCESS

Data Picker Experiments

A variety of Data Picker layouts are created to arrive at a layout. These layouts differed in their organisation of glyphs. The different organisations are the result of experimenting with the different neural network models available through the smartgrid script.

Below are some examples from the experiments:

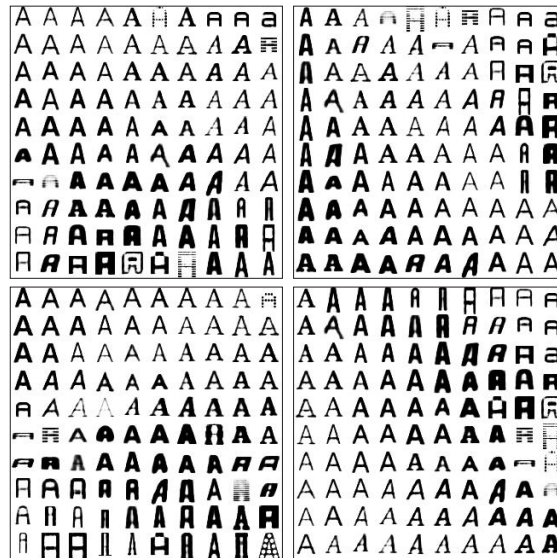


Figure 15: A selection of Data Picker experiments.

The selected layout (bottom right in Figure 15) was chosen for its strong sense of organisation.

SPACESHEETS: DESIGN EXPERIMENTATION IN LATENT SPACE

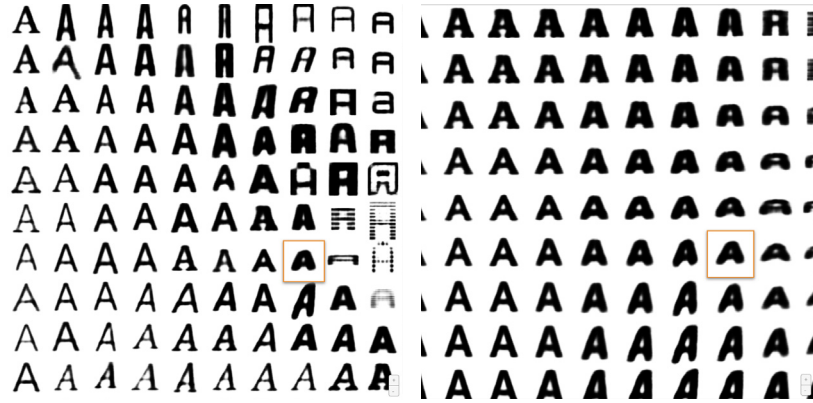


Figure 16: Intermediate variable interpolations as the Data Picker is zoomed in upon.

Summary of Features

Data Picker

- Has a grid of font reconstructions.
- Clicking on it populates the selected cell in the spreadsheet.
- Users can zoom in on the grid to produce intermediate variables.
- Has zoom-in and zoom-out buttons.

Spreadsheet

- Cells support latent variables.
- A bar at the top of the spreadsheet which displays the formula of the selected cell.

DESIGN PROCESS

- Operations for latent variables:
 - AVERAGE
 - LERP
 - MINUS
 - SUM
 - DIST
 - SLIDER
 - RANDFONT
- Users can click to reference a cell.
- Cell highlighting.

Workflows

As the application was tested, several useful workflows emerged. These are motivated by the techniques used to sample generative networks identified in the background section.

Font Tweening

This enables users to interpolate between two fonts through a slider cell.

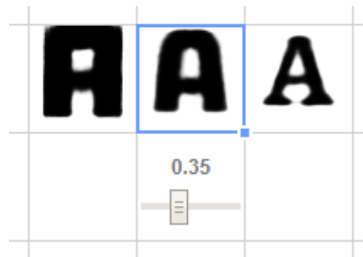


Figure 17: Tweening between two glyphs using a slider element.

SPACESHEETS: DESIGN EXPERIMENTATION IN LATENT SPACE

Applying Attribute Vectors

Attribute vectors can be isolated first by evaluating the difference between two variables using the *MINUS* operator. This difference can then be applied to other variables to transform it into looking more like the attribute.






bold glyph (a)	light glyph (b)	bold vector (a - b)	input (x)	bolded input (x + (a - b))
				

Figure 18: Isolating and applying a bold vector in the SpaceSheet.

Fine tweaking Data Picker references

Similar variations of a given glyph can be made by tracing back to its Data Picker references and picking references which neighbour it. Minor variations can be made by zooming into the reference and picking subdivisions around its general region. This is particularly effective for finding clearer, more polished alternatives for a given glyph.

User Testing

John the Spreadsheet Expert

John is a Data Analyst who uses spreadsheets on a daily basis at his work, where he checks data quality and works with an in-house designer to create data visualisations.

Besides his work, he has a personal project for which he has bootstrapped website and logo designs for. This has been done through unconventional means - using *Microsoft Powerpoint*, due to the cost of licensing specialist design tools.

He describes the process of selecting fonts as a search for fonts with a balance between “*a particular style*”, and “*something readable*”.

DESIGN PROCESS

George the Interface Designer

George is an Interface Designer who deals with digital designs in his daily practice. This involves the design of websites and user interfaces.

He is a font aficionado, stating that he *“absolutely loves”* certain typefaces. So much so, that he once attempted to create a typeface. He describes that the process as one that *“takes forever”*: *“I had to take all the vectors from Adobe Illustrator, into specific font making software, had to resize them all and it didn’t fit, then I had to generate the font at the end, and this was before even doing any kerning, or spacing, or different weights...”*

He describes that his preference for fonts: *“Generally it comes down to legibility. I don’t really like decorative fonts... usually, it’s geometric, sans serif, grotesque fonts”*. But notes that he tends to pick *“more interesting, more fun, more humanist”* fonts when designing a poster.

He describes his process for selecting fonts as *“basically experimentation”*, cycling through a range of fonts to *“see what looks good”*.

He uses spreadsheets casually: *“Usually it’ll be a budget for something..basically using it as a glorified calculator”*.

Results

Two user tests were conducted for this version of the application. As the operations were implicit within the application at the time, each participant was provided with a printed sheet which documented the operations available and their required parameters.

For a detailed description of the testing process, refer to the methodology section.

The user testing revealed that the tool made it *“easy to explore”* solutions, but *“challenging to find a specific thing that you’re after”*. This is attributed to the different mode of working, which requires a different mental model than that of other design tools. George agrees, stating that, *“I think my mental model of how the vectors and space work is quite unclear”*.

SPACESHEETS: DESIGN EXPERIMENTATION IN LATENT SPACE

This confusion is largely related to the user intuition of the *SUM* and *MINUS* operations. Confusion around these operations made it challenging for users to isolate and apply attribute vectors.

George mentioned that *“we don’t usually go around adding abstract concepts”*. The results of these operations often resulted in illegible reconstructions, as they would be samples from improbable locations in the latent space.

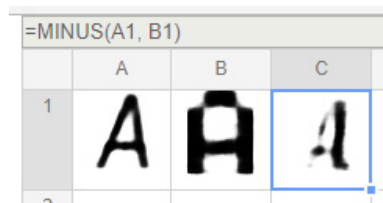


Figure 19: Illegible representation of an attribute vector.

Additionally, there were moments where the user would confuse pixel representations with the more abstract representations. This was revealed in John’s mention of *“overlapping”* features, and questioning the results of a subtraction: *“Why is it that when you subtract a thicker font from a thinner font, you’re not left with an empty image?”*.

As a result of this, the *LERP* operation was used frequently by both users. Transitioning between two letters was intuitive. Users were able to find most of their search targets using this operator. George made the connection between the *LERP* operator and slider and described it to be *“the funnest bit”* of the application.

There were a few issues with the Data Picker. It wasn’t immediately obvious that it was zoomable, and it was assumed that the variables within it are exhaustive of the tool’s possibilities.

John mentions that the subdivisions enabled flexibility: *“being able to zoom down”* to get *“slight variations gives you a quite a bit of flexibility”*. He also explains that the selection of variables within it are strong determinants of what attributes can be isolated.

DESIGN PROCESS

Despite the shortcomings of the tool, the users were engaged, and the spreadsheet interface enabled them to explore flexibly. It made it *“very easy to track things”*. The ability to express relationships between the variables was described to be *“very creative in the sense that you are always riffing on things constantly”*.

George idealises a *“mass LERP” which consists of multiple sliders and LERP setups, to which he can “lerp away to [his] heart’s content”*.

A workflow emerged as John attempted to find search targets by sequentially interpolating between variables. This is reflected in a layout which he describes as resembling a *“decision tree”*. He describes his thought process: *“What I was trying to do was I was trying to find two fonts that could come together or broken apart to get to the font I was looking for... once I had aspects of the two, I would come down and be like “do I want to make any more adjustments?”...if I do, then I would bring on another font. It’s kind of like a decision tree. Do these two work together? Yes. Ok, bring the product down and make changes to the product. If they don’t work together, start again from scratch with different fonts and have another go”*.



Figure 20: John’s decision tree layout.

George relates the tool to his experience building a font and suggests that *“what I can get out of it is so worth the effort...compared to making a font by hand, it is infinitely faster”*. He adds that it would be useful to view the whole font, perhaps with a custom text field — *“Sometimes when I’m designing something, I’ll have a key thing[sentence] in mind and I want to find a font for it... I’ll try fonts just specifically for that word”*.

SPACESHEETS: DESIGN EXPERIMENTATION IN LATENT SPACE

Dimension	Agreement Statement	John	George	Average
Enjoyment	I would be happy to use this tool on a regular basis.	6	6	6
	I enjoyed using the tool.	8	9	8.5
Exploration	It was easy for me to explore many different ideas, options, designs, or outcomes, using this tool.	7	8	7.5
	The tool was helpful in allowing me to track different ideas, outcomes, or possibilities.	6	8	7
Expressiveness	I was able to be very creative while doing the activity inside this tool.	9	8	8.5
	The tool allowed me to be very expressive.	8	5	6.5
Immersion	My attention was fully tuned to the activity, and I forgot about the tool that I was using.	8	5	6.5
	I became so absorbed in the activity that I forgot about the tool that I was using.	8	5	6.5
Results Worth Effort	I was satisfied with what I got out of the tool.	7	9	8
	What I was able to produce was worth the effort I had to exert to produce it.	7	9	8

Table 3: John and George's Creativity Support Index Agreement Scores.

The user ratings on the Creativity Support Index Agreement Statements are consistently high, apart from *Immersion*. My intuition is that for a user to be immersed, the user first needs to be comfortable and familiar with the tool. This is supported by the results in that John, rated immersion much higher than George, who is less versed with spreadsheet interfaces.

DESIGN PROCESS

Evaluation

This version of the tool confirms that the interactivity of the spreadsheet interface is a flexible tool for exploration in latent space. However, user testing indicates that the tool offers a new and unfamiliar mode of working with designs.

It is unfamiliar in two ways: spreadsheet tools are not part of the conventional designer's toolkit, and operations within latent space is an ambiguous concept to grasp.

Two things are required to improve the fit of the tool:

1. Modifying the conventional spreadsheet interface to be a better fit as a tool for design experimentation.
2. Enabling users to form an appropriate model of how to make sense of latent spaces through the operations.

Additionally, a lot of the information required to use the tool is currently implicit — the operations and their parameters. This should be made explicit in the tool to enable users to use the tool without any external aids, or prior knowledge.

SPACESHEETS: DESIGN EXPERIMENTATION IN LATENT SPACE

VERSION TWO

The developments in this version include the introduction of two interface elements: the *Operation Bar* and a *Font Drawer*. These elements are designed in response to the feedback gathered from the first round of user testing.

Operation Bar

The *Operation Bar* is a strip of buttons — one for each operation — which can be clicked to populate a selected cell with its respective operation. Additionally, hovering over the buttons highlights the cells which will be populated if the button were clicked. This is designed to address many of the application's limitations by:

- Making the operations explicit.
- Enabling users to populate cells more naturally and efficiently.
- Aiding user learning through suggestions of operations.

The buttons have two states: suggested and regular. These states are indicated by a change in colour: orange when suggested, and grey when regular. Buttons are suggested if the selected cells are valid parameters for the button's operation.

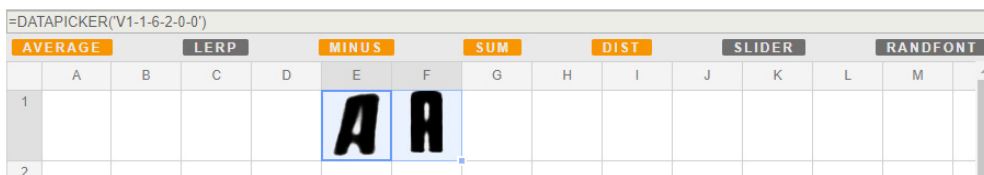


Figure 21: Operations are suggested based on the selection.

DESIGN PROCESS

For example, if two cells with latent variables are selected, the *SUM* operation button will become suggested. Clicking the operation will sum the variables in an adjacent cell by using the selected cells as parameters to the operator.



Figure 22: Suggested cells to be populated are highlighted as the user hovers over a suggested operation.

The logic to suggest cells is referred to as *SmartFill*. There are four SmartFill variants: *Double*, *Group*, *Corners*, *Fill*. These variants are created to support the different requirements of different operations.

The *Double SmartFill* is used on the operations which require two parameters: *SUM*, *MINUS*, *DIST*. The SmartFill is suggested when there are exactly two values (which are not diagonal to each other) selected in the spreadsheet. If empty cells are present in the selection, the SmartFill populating the first empty cell in the selection. Otherwise, it suggests to populate the first cell outside the selection, relative to the layout of the selection — horizontal and vertical layouts are supported.

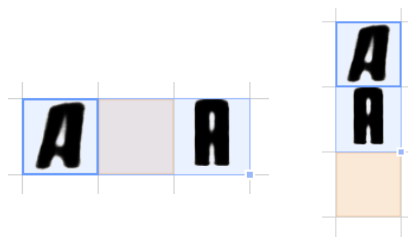


Figure 23: Variety of Double SmartFill configurations.

SPACESHEETS: DESIGN EXPERIMENTATION IN LATENT SPACE

The *Group SmartFill* is used on the *AVERAGE* operator. The SmartFill is suggested when multiple cells are selected, and the selection contains two or more values. The first cell outside the top right corner of the selection is selected if there is a cell present. Otherwise, cells from the other corners are suggested.

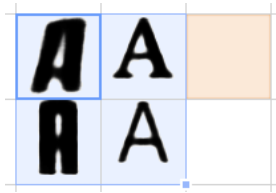


Figure 24: Group SmartFill.

The *Corner SmartFill* is used on the *LERP* operator. It is suggested when there are values at each corner of the selection. The cells that are suggested to be populated are the cells between the corner cells. This enables users to interpolate between points and across multiple cells efficiently.

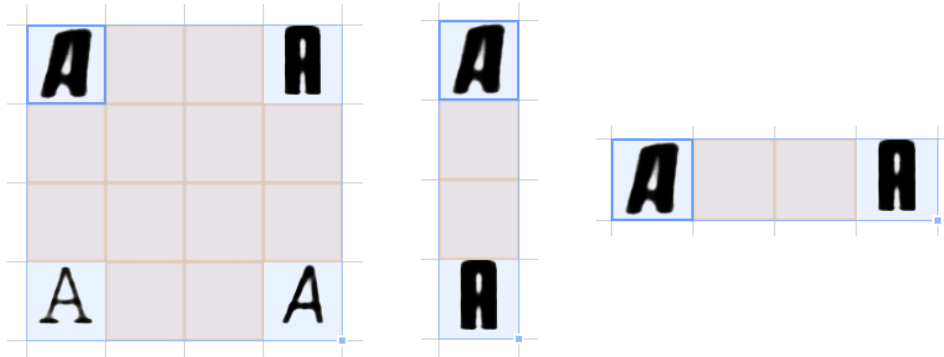


Figure 25: Miscellaneous Corner SmartFills.

DESIGN PROCESS

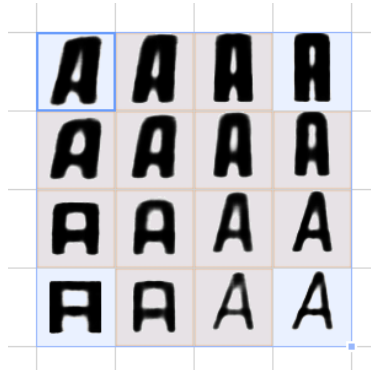


Figure 26: A grid of interpolations created with the Group SmartFill.

The *Fill SmartFill* is used on the *SLIDER* and *RANDFONT* operators. It simply suggests filling all the cells within a selection.

Font Drawer

The *Font Drawer* is a section at the bottom of the spreadsheet which holds a list of *Font Samples* referenced from the spreadsheet. This has been designed to enable users to test a font on different letters, or custom strings of text.

Font Samples are elements which reference cells within the spreadsheet. These elements render custom strings of text, defined in an input field at the top of the Font Drawer. Cell references can be assigned textually via an input field or by clicking on the '+' button to reference the currently selected cell in the spreadsheet.

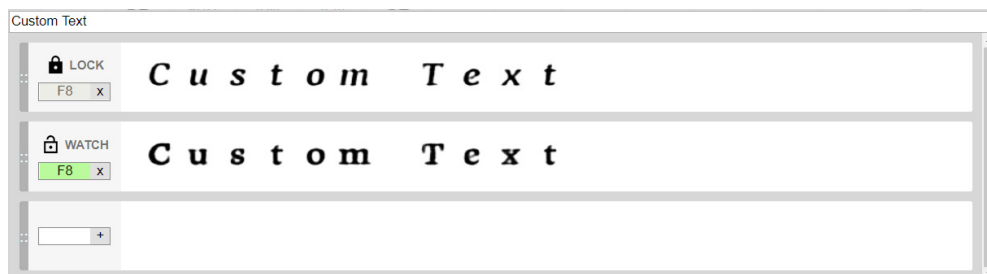


Figure 27: Font drawer with stored samples from the spreadsheet.

SPACESHEETS: DESIGN EXPERIMENTATION IN LATENT SPACE

Font Samples have two states: *watch* and *locked*. These can be toggled via a button click. When a Font Sample is in its watch state, it updates the rendered font as the value of its corresponding cell changes in the spreadsheet. Locked cells enable the user to prevent the font from updating as its reference is modified.

Data Picker Improvements

User feedback on the Data Picker revealed that it falsely indicates that it is exhaustive of the possibilities of the space and that it isn't obviously zoomable. To deal with this, several Data Picker layouts are created and made accessible through tabs at the top of the Data Picker element.

These layouts are created to serve themes or different classes of fonts. This is intended to showcase the diversity of images that can be generated by the space, as well as to act as an organisation tool.

One of these layouts is the standard layout: a collection of fonts of similar shapes and sizes. It is curated to enable users to isolate single attributes from fonts easily.

Additional user interface elements are added as overlays to the Data Picker to make it more obvious that it is a zoomable element. The zoom percentage and bounds of the viewport are displayed at the bottom right corner of the Data Picker. Additionally, a '100%' button has been added to enable users to reset back to the initial level of zoom easily.

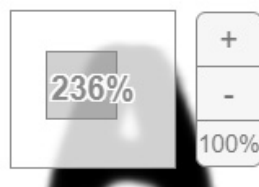


Figure 28: Information overlay on the Data Picker
illustrating the viewport position and scale.

DESIGN PROCESS

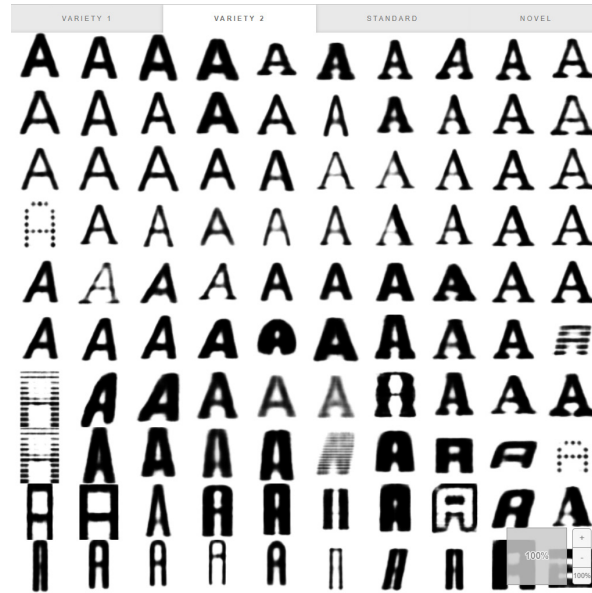


Figure 29: Data Picker with multiple layouts and a UI overlay.



Figure 30: Similar fonts with minimal differences have been selected for the Standard layout.

SPACESHEETS: DESIGN EXPERIMENTATION IN LATENT SPACE

Summary of Modifications

Operation Bar

- A strip of buttons which can be used to populate cells.
- Can infer cells to be populated based on the selected cells.

Font Drawer

- Enables users to store references to cells in the spreadsheet.
- Renders custom strings of text.

Data Picker

- Added various layouts to choose from.
- Added additional information overlays.

DESIGN PROCESS

Experiment: Constructed Design Space

As an experiment, an alternative version of the application is created to operate within a constructed design space instead of a latent space. Instead of decoding the variables through the generative model, numbers within the variables are used as parameters for a drawing. A variety of sketches were attempted: faces, constellations, rotated rectangles.



Figure 31: Data Picker of face drawings.

Operations between the encoded sketches using the spreadsheet are intuitive. A simple sketch with basic geometry could be used as an example for users to understand the spreadsheet interface.

SPACESHEETS: DESIGN EXPERIMENTATION IN LATENT SPACE

User Testing

Paul the Architecture Postgraduate

Paul is a postgraduate architecture student who has designed a sustainable structural system for architecture to support a “*circular economy*”. The project involved a digital workflow for him to model, evaluate and manufacture parts for a full-scale build of a structure. This workflow involved a sequence of 3D modelling software including *SketchUp*, *Rhino with Grasshopper* and *Adobe Illustrator*. 3D sheet modelling was done in SketchUp, evaluated in Grasshopper and then imported into Illustrator to create shapes to be laser cut.

He is an expert user of spreadsheets, stating that he “*used spreadsheets for everything*” in the project — to evaluate his designs and even to plan out his tasks. This is due to its ability to organise: “*I knew that I could go into this one spreadsheet and I would have the plan, the budget, the references, the contacts, the evaluation of the systems, the costing, it was all there*”.

He is minimal with his font selection, having a few versatile favourites: “*I love Avenir Next and Myriad Pro, and Times. That’s all I ever use. I don’t go out looking for new fonts, I like to be conservative*”.

Results

In this round of user testing, Paul is shown the font version of the application initially, followed by the version with drawn faces.

The user testing revealed that the Operation Bar made the system and its operations more intuitive and efficient. It made the list of operations available explicit, and also previewed the cells that it would modify. It enabled the user to start working and experimenting easily. However, there were occurrences where specific sequences of operations instantiated within the bar would overwrite cells and result in circular references.

The effect of the operations in latent space was unintuitive to Paul. The results of the *SUM* and *MINUS* operations were unexpected. This is again, due to the confusion between pixel and abstract level representations, as well as the

DESIGN PROCESS

representation of attribute vectors: “...what I’d expect is that if you had a fat font, and took away a skinny font, you’d have something in between. But instead, you’ve got a fatter font”. He found operating between the faces in constructed space to be more intuitive, to the extent that it felt like a different application: “This is less unexpected. This one is more defined...the other one is almost a different process”.

The resolution of the outputs was important for Paul. When using the version with the drawn faces, he describes: “What I like about this, more than the fonts, is that they look complete. Whereas with the text, it was always blurry. This does a lot more for me. This is more tangible”. He explains the value of the tool in generating a set of designs, giving the example of having to “design a scene with 1000 orcs”.

When asked about the spreadsheet as a design tool, Paul described it as “very confrontational, in that it is fixed”. He then affirms the need for the separate sections in the application “...I want the data, the play space, and then the output field”. This is achieved by the Data Picker, spreadsheet, and Font Drawer respectively.

Dimension	Agreement Statement	Score
Enjoyment	I would be happy to use this tool on a regular basis.	7
	I enjoyed using the tool.	8
Exploration	It was easy for me to explore many different ideas, options, designs, or outcomes, using this tool.	7
	The tool was helpful in allowing me to track different ideas, outcomes, or possibilities.	5
Expressiveness	I was able to be very creative while doing the activity inside this tool.	5
	The tool allowed me to be very expressive.	5
Immersion	My attention was fully tuned to the activity, and I forgot about the tool that I was using.	2
	I became so absorbed in the activity that I forgot about the tool that I was using.	2
Results Worth Effort	I was satisfied with what I got out of the tool.	5
	What I was able to produce was worth the effort I had to exert to produce it.	8.5

Table 4: Paul’s Creativity Support Index Agreement Scores.

SPACESHEETS: DESIGN EXPERIMENTATION IN LATENT SPACE

Similarly to the results in the first round of user testing, it was found that the tool was fun to use and good for exploring, but hard to control. This is again due to the unpredictable results of the operations in latent space.

The unpredictability correlates to the low agreement scores for *Creativity* and *Expressiveness*. Paul suggests that being able to express glyphs by sketching would enable him to be more creative.

In response to *“the tool was helpful in allowing me to track different ideas, outcomes, or possibilities”*, he mentions that it was hard to *“understand the connections between cells, whereas in Grasshopper you can easily delete and modify junctions”*.

Evaluation

This round of user testing reinforced the opinion of the tool being *“hard to control”*. This is again, largely due to the user’s mental model of the latent space operations. The user’s feedback on the face version of the application revealed that operations in the constructed space were more intuitive than abstract operations in latent space.

With regards to the interface, the Font Drawer was useful in providing an *“output field”* for a design. Improvements could be made in making the relationships between cells explicit — Paul stated that it was hard to keep track of relationships between cells.

For the next iteration:

1. Devise methods to aid users in developing a mental model for latent space operations.
2. Reveal the relationships between cells explicitly.

DESIGN PROCESS

VERSION THREE

This version aims to assist the user in building their intuition on latent space operations, as well as to make the relationships between cells more explicit. This is attempted by the introduction of a demonstration sheet, and changing the behaviour of cell highlighting.

Demo Sheet

User testing revealed that the *SUM* and *MINUS* operations were not intuitive. This results in an inability to control specific dimensions of a glyph, as it inhibits the process of isolating and applying attribute vectors.

As a quick solution to help users build intuition around attribute vectors, a sheet is predefined to demonstrate the process of isolating these attributes.

The demo sheet is broken into two parts: isolating the attribute vectors and applying the attribute vectors.















has property				
doesn't have property				
difference				
scaled difference				
input	boldness	serif strength	italic strength	transformed
	0.00 	0.00 	0.10 	

Figure 32: Demo sheet which isolates and applies attribute vectors to a variable.

SPACESHEETS: DESIGN EXPERIMENTATION IN LATENT SPACE

Three properties are isolated: boldness, serif strength and italic strength. This involves picking a glyph with the property, a glyph without the property and finding its difference. These values are labelled in adjacent cells to describe the process to the user.

The application of the attribute vectors is controlled by a series of sliders which define the amount of the attribute vector to be applied. For this, a *MUL* (multiply) operator is implemented to scale the attribute vector. An input cell is labelled, and glyphs in this field are transformed by a series of attribute vectors into a labelled output cell.

The *DIST* operation is removed from the Operation Bar to make space for the *MUL* operator.

DESIGN PROCESS

Explicit Cell Relationships

The behaviour of highlighting cell references is modified to be more explicit. This is in response to the user feedback which indicated that it was hard to understand the relationships between cells.

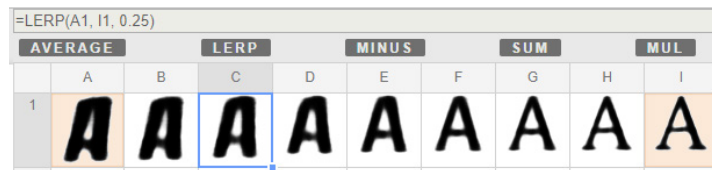


Figure 33: Cell references within the selected cell are highlighted.

Previously, cell references would only be highlighted if the user was editing a cell value. This has been modified so that the selection of a cell also highlights the cells it references.

Summary of Modifications

Demo Sheet

- Added a *MUL* operator to scale attribute vectors.
- A template which isolates and applies bold, serif and italic vectors.

Explicit Cell Relationships

- Selecting a cell highlights the other cells it references.

SPACESHEETS: DESIGN EXPERIMENTATION IN LATENT SPACE

User Testing

Ringo the Parametric Architecture Designer

Ringo is a postgraduate architecture student, having three years of parametric and computational design experience. He is familiar with *Rhino with Grasshopper*. He describes his process of designing architecture as “*derivative*”, in that he derives forms from strict parameters. This is explained in how he uses parametric design tools: “*I give it strict parameters that I want it to work around and what results are that I have some control but it will more or less give me something that is derived from parameters*”.

He uses spreadsheets when necessary but is not proficient in it. He describes that this is due to its standardised format: “*For me, the use of spreadsheets is as an application for something else. It’s a common format, it’s a standard format that can go into other applications*”.

He has clear preferences for fonts, and favours “*clean, sharp, crisp, no extra bits, mono*”. He elaborates on the importance of specific proportions in fonts: “*The most important thing is the proportion of it if parts of it are too large or too small, it throws the whole font off*”.

Results

This round of user testing showed that the user had started to build an intuition for latent space operations, reinforced the effect of several existing issues, and revealed a “*crossword puzzle*” method of working with the interface.

Ringo exhibited an intuition of latent space operations, providing reasons why certain operations would not be helpful to achieve his goal. When applying attribute vectors: “*You can’t add that because you’ll introduce more elements*”. However, this intuition was fairly rudimentary. When presented with the demo sheet, it took Ringo some time to understand the effect of the attribute vectors. This was due to the uncommon terminology and layout of the sheet. “*Has property*” and “*doesn’t have property*” were unnatural terms to the user.

The effects of the *SUM* and *MINUS* operators were again unexpected. It was interesting to note that although Ringo found the *SUM* results unexpected, it was

DESIGN PROCESS

amusing to him. He uses it to create abstract forms: *“The SUM tool is not something I am using to try to find a font, but I am using to create something that’s quite abstract”*.

Using *LERP* across multiple cells was the most common operation used by Ringo to reach a search target. When asked why, he explains: *“I can pick out a certain element (within a font) with a property I want, then I can pick out a second element with some properties that I want then LERP. I don’t know if it’s 60% of this one and 40% of the other one. LERP lets me figure it out”*.

Additionally, he notes that using *LERP* across cells is more useful and intuitive than using a slider. It enabled him to view all the iterations visually at once for better evaluations. He explains: *“For me, I wasn’t sure how to integrate the slider and I’m not great with spreadsheets. The LERP button is much more intuitive... it’s more visual. Now I’ve got all these iterations, I can say ‘that’s what I want’ or I have that moment where I say, ‘Actually, maybe that one is just a little bit better’...it’s very hard to compare things with the slider”*.

He uses this technique extensively to find search targets in a “crossword puzzle” layout. This layout would start with an interpolation between two fonts. The most suitable candidate across this interpolation would be selected as the basis for another interpolation orthogonal to the first interpolation. This process repeats until he arrives at a satisfactory result.



Figure 34: Ringo’s “crossword puzzle” layout.

Ringo supports the cell highlighting, describing it as *“a way to keep things manageable”*.

SPACESHEETS: DESIGN EXPERIMENTATION IN LATENT SPACE

Dimension	Agreement Statement	Score
Enjoyment	I would be happy to use this tool on a regular basis.	8
	I enjoyed using the tool.	7
Exploration	It was easy for me to explore many different ideas, options, designs, or outcomes, using this tool.	8
	The tool was helpful in allowing me to track different ideas, outcomes, or possibilities.	9
Expressiveness	I was able to be very creative while doing the activity inside this tool.	6
	The tool allowed me to be very expressive.	6
Immersion	My attention was fully tuned to the activity, and I forgot about the tool that I was using.	5
	I became so absorbed in the activity that I forgot about the tool that I was using.	6
Results Worth Effort	I was satisfied with what I got out of the tool.	9
	What I was able to produce was worth the effort I had to exert to produce it.	7

Table 5: Ringo’s Creativity Support Index Agreement Scores.

Ringo scored the tool highly on all dimensions in the Creativity Support Index Agreement Statements, apart from *Expressiveness* and *Immersion*.

Expressiveness was scored lowly due to the “wacky” and unexpected results from the *ADD* operator.

The low scores in *Immersion* could be due to the frequent occurrence of circular references when modifying the demonstration sheet. In cases where circular references are entered, the application crashes — logic to prevent this has not been implemented due to time constraints. He describes that he was “scared to try” an operator after the application crashed from a circular reference.

DESIGN PROCESS

Evaluation

This round of user testing indicated that a demonstration sheet was useful in building user intuition of latent space operations.

Despite this, the representation of the results of the *ADD* and *SUBTRACT* operators were still unexpected and confused the user.

Additionally, the ability to create interpolations across cells was the most common feature used. This feature was used to arrive at most search targets, in a layout which resembled a “*crossword puzzle*”.

The application had not been implemented to detect and deal with circular references. This made the application crash, making the user less immersed using the tool.

Overall, Ringo had an enjoyable experience. He states that: “*I was so damn happy when I got that close to those three fonts that you asked me to get to*”.

SPACESHEETS: DESIGN EXPERIMENTATION IN LATENT SPACE

RESULTS

RESULTS

The results of user testing reinforced the use of the tool as a new mode of working with designs. It enabled users to explore design possibilities from a top-down approach, by deriving meaning and navigating within a preconstructed model, rather than requiring a model to be constructed from the bottom-up.

Dimension	Agreement Statement	John	George	Paul	Ringo	Average
Enjoyment	I would be happy to use this tool on a regular basis.	6	6	7	8	6.75
	I enjoyed using the tool.	8	9	8	7	8.5
Exploration	It was easy for me to explore many different ideas, options, designs, or outcomes, using this tool.	7	8	7	8	7.5
	The tool was helpful in allowing me to track different ideas, outcomes, or possibilities.	6	8	5	9	7
Expressiveness	I was able to be very creative while doing the activity inside this tool.	9	8	5	6	7
	The tool allowed me to be very expressive.	8	5	5	6	6
Immersion	My attention was fully tuned to the activity, and I forgot about the tool that I was using.	8	5	2	5	5
	I became so absorbed in the activity that I forgot about the tool that I was using.	8	5	2	6	5.25
Results Worth Effort	I was satisfied with what I got out of the tool.	7	9	5	9	7.50
	What I was able to produce was worth the effort I had to exert to produce it.	7	9	8.5	7	7.88

Table 6: Creativity Support Index Agreement Scores from all User Testing Sessions.

For this reason, the tool scored highly in the dimensions of *Enjoyment*, *Exploration* and *Results Worth Effort* on the Creativity Support Index Agreement scales. George compares his experience of the tool with his experience creating a typeface from a bottom-up approach: “*what I can get out of it is so worth the effort...compared to making a font by hand, it is infinitely faster*”.

SPACESHEETS: DESIGN EXPERIMENTATION IN LATENT SPACE

Additionally, providing a model to operate within enabled non-designers to explore feasible design possibilities without requiring the skill needed to operate parametric and direct manipulation design tools. This can be seen in John, the spreadsheet expert's high scoring of *Expressiveness*, in contrast to the designers' low scores for that dimension. When asked how he would otherwise explore typefaces, John replied saying that he wouldn't be able to.

User testing also revealed that operating within latent space was a hard concept to grasp. This was due to the novelty of adding and subtracting abstract concepts, and the expectation of pixel level differences instead of abstract level differences.

This ambiguity had a strong influence on the operations users were comfortable in attempting, which in turn affected what they were able to achieve with the tool. It discouraged users from attempting to sample the latent space using attribute vectors.

Instead, users relied heavily on creating interpolations to find search targets. The act of visually transitioning from one glyph to another was much more intuitive. Although interpolation was found to be effective to arrive at most search targets, users were not able to achieve fine attribute-level control in their experiments. This is reflected in the low agreement scores for *Expressiveness* and *Immersion*.

A developed understanding of attribute vectors is crucial to use the tool to its full potential. Two solutions have been proposed to build this understanding: visually distinguishing between variables and attribute vectors, and creating demonstration sheets.

Attribute vectors are the result of the difference between two latent variables. It represents an operation to be applied to a variable, rather than a variable itself. In the tool, this is achieved through the *MINUS* operation. The representation of these attribute vectors was a source of confusion, as they often resulted in illegible images of glyphs. Creating a visual distinction between these two entities may be useful in establishing the different behaviours between the values. This has not been implemented in the course of the research but seems to be a promising improvement.

RESULTS

A demonstration sheet was devised to illustrate the process of isolating and applying attribute vectors. This was shown to be useful in developing a rudimentary user understanding. Further efforts in refining the sheet and developing several variants could prove to be more effective in building user understanding.

The demonstration sheet also suggested an alternative mode of working with the interface: as a tool to define interfaces. This enables new users to explore within meaningful attribute-level constraints defined in the template. Due to the flexibility of the spreadsheet interface, users are able to customise these interfaces as they develop a better understanding of the tool and how to use it.

In summary, the tool presents a novel top-down approach to design experimentation. This was reported to be more efficient, and supportive of exploration than bottom-up approaches. Additionally, it enabled non-designers the ability to experiment with designs. However, user observation revealed that an understanding of new concepts and skills is required for the tool to be used to its full potential.

Interpolation was found to be the most intuitive method to arrive at search targets, however, this limited the amount of control users had over their experiments. An understanding of sampling using attribute vectors is required to enable finer control in design experiments. Initial efforts to support user understanding were shown to be promising in building user intuition.

SPACESHEETS: DESIGN EXPERIMENTATION IN LATENT SPACE

CONCLUSION

CONCLUSION

Overall this work has shown the potential of the use of latent spaces in computational design tools to enable more effective and meaningful design experimentation. This investigation was facilitated through the design and evaluation of the *SpaceSheet*, a general purpose spreadsheet interface which enables users to sample latent spaces.

User observation and evaluation of the SpaceSheet revealed that latent space exploration enabled a different approach to work with designs. It offered a top-down method to experiment with designs, which enables users to express design intent through design principles implicit within the latent space. This method of working was reported to be more supportive of design exploration, more efficient, and capable of enabling non-designers to explore design possibilities.

Unsurprisingly, the new approach required new skills and intuition to be used to its full effect. User testing revealed that a lack of knowledge in deriving and applying attribute vectors from latent space limited user's expressivity and control over their experiments. Interpolation was found to be much more intuitive and effective to arrive at most search targets. However, interpolation did not enable the user to ability to express attribute-level operations.

In light of this, the ability to derive meaning from latent spaces can be considered an expertise independent of design experimentation. The expertise required goes beyond the understanding of the techniques required, but also an intuition of the properties inherent in latent spaces. An example of this is the ability to minimise the unwanted effects of correlated labels in attribute vectors. These results yield different results and its effect is subject to the intent of the user.

The SpaceSheet acts as a flexible, low-level interface which provides the means required for users to build their intuition of sampling latent spaces. The construction of a demonstration sheet, which wraps low-level functions into higher-level controls shows promise as a starting point to build user intuition.

The limitations of the tool became apparent when users attempted to express concrete transformations to designs in latent space. These often resulted in unfavourable results.

SPACESHEETS: DESIGN EXPERIMENTATION IN LATENT SPACE

Just as low-level deterministic methods are an ill fit to express abstract design intent, the high-level probabilism of sampling latent space is an ill fit to express concrete design intent. For example, moving the position of an image can be accomplished certainly using conventional means. Expressing this operation in latent space provides redundant uncertainty. It is worth noting however, that this uncertainty has been reported to be serendipitous on occasion. These moments occur when the results do not match the user's expectations — due to distortions and artifacts in the reconstruction — but is still pleasing.

SpaceSheets explores the potential of latent spaces to be used as a tool for design experimentation. The research finds it to enable a novel method to work with designs, which supports more efficient, high-level design experimentation to designers and non-designers alike. The user's intuition and ability to derive meaning from latent spaces is fundamental in using the tool to conduct design experiments with a fine level of control. This intuition can be considered a skill, which can be developed through continued experience with the flexible, low-level interface provided by the SpaceSheet.

Although latent spaces enable designers to express more meaningful design operations, it provides redundant uncertainty for concrete design operations. It is with this understanding that latent spaces are best considered as a complementary new primitive to build smarter design tools.

LIMITATIONS AND FUTURE WORK

User testing was limited to first-time users of the tool. This was useful in identifying usability issues, thoughts, and opinions on the tool, but not indicative of its potential when used by experienced users. Longitudinal case studies, with recurring users, would have been useful in understanding a) the ease of acquiring sufficient intuition on sampling latent spaces and b) the added benefits of having attribute-level control in design experiments.

Although the tool is built to be general purpose, only a model of fonts was implemented. This was due to the lack of available pre-trained models suitable for the course of the research. User exploration of various models could reveal additional insights to create more generalised findings.

CONCLUSION

Taking the viewpoint that sampling latent space is an expertise that can be developed, it is most important that additional mechanisms are implemented to reveal the implicit properties of the latent space.

For example, variables are decoded into images and provide no indication of its 'location' in latent space. The images of two decoded variables may look similar, but be distant from each other in latent space. A supplementary representation could be provided to make its location more explicit.

The design experimentation afforded by the SpaceSheet is currently limited to a single design component, in this case, fonts. The integration of multiple sheets, as in conventional spreadsheets enables users to refer to data between sheets. This is perhaps a solution to enable users to experiment with compositions of design components. Multiple sheets which support various models could be implemented, and these can be referred to in a separate section to create compositions of the elements which can be experimented with.

The ability of the SpaceSheet to enable non-designers to explore design possibilities could be used as a tool for clients to communicate more effectively with designers. This could be facilitated through a collaborative online spreadsheet.

The ability to sketch a glyph as to use as a variable was both considered and commonly requested in user feedback. This had not been implemented due to time constraints. Sketching is a more natural way to express designs and enables users to explore using a wider range of variables than that presented in the Data Picker.

SPACESHEETS: DESIGN EXPERIMENTATION IN LATENT SPACE

BIBLIOGRAPHY

BIBLIOGRAPHY

Alexander, C. (1964). *Notes on the Synthesis of Form* (United States). Harvard University Press.

Bernhardsson, E. (2018). *deep-fonts: Generate fonts using deep learning*. Python. Retrieved from <https://github.com/erikbern/deep-fonts> (Original work published 2015)

Carter, S., & Nielsen, M. (2017). *Using Artificial Intelligence to Augment Human Intelligence*. Distill, 2(12), e9. <https://doi.org/10.23915/distill.00009>

Cherry, E., & Latulipe, C. (2014). *Quantifying the Creativity Support of Digital Tools Through the Creativity Support Index*. ACM Trans. Computer-Human Interaction, 21(4), 21:1–21:25. <https://doi.org/10.1145/2617588>

Cross, N. (1982). *Designerly Ways of Knowing*. Design Studies, 3(4), 221–227.

Dorst, K., & Dijkhuis, J. (1995). Comparing paradigms for describing design activity. Design Studies, 16(2), 261–274. [https://doi.org/10.1016/0142-694X\(94\)00012-3](https://doi.org/10.1016/0142-694X(94)00012-3)

Handsontable - JavaScript Spreadsheet Component For Web Apps. (n.d.). Retrieved 6 July 2018, from <https://handsontable.com/>

Hansmeyer, M. (2012). *Building unimaginable shapes*. Retrieved from https://www.ted.com/talks/michael_hansmeyer_building_unimaginable_shapes

Hebron, P. (2016). *Machine Learning for Designers*. CA, USA: O'Reilly Media Inc. Retrieved from <https://www.safaribooksonline.com/library/view/machine-learning-for/9781491971444/>

Hebron, P. (2017, April 26). *Rethinking Design Tools in the Age of Machine Learning*. Retrieved 27 June 2018, from <https://medium.com/artists-and-machine-intelligence/rethinking-design-tools-in-the-age-of-machine-learning-369f3f07ab6c>

Hewett, T., Czerwinski, M., Terry, M., Nunamaker, J., Candy, L., Kules, B., & Sylvan, E. (2005, September). *Creativity Support Tool Evaluation Methods and*

SPACESHEETS: DESIGN EXPERIMENTATION IN LATENT SPACE

Metrics. Retrieved 25 June 2017, from http://s3.amazonaws.com/academia.edu.documents/29985520/nsfcreativitybook_final.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1498429183&Signature=IMRj1yXFOq9gEw13OwcbfW6%2FojQ%3D&response-content-disposition=inline%3B%20filename%3DCreativity_support_tool_evaluation_metho.pdf#page=13

Larkin, J. H., & Simon, H. A. (1987). *Why a diagram is (sometimes) worth ten thousand words*. *Cognitive Science*, 11(1), 65–100.

Larsen, A. B. L., Sønderby, S. K., Larochelle, H., & Winther, O. (2015). *Autoencoding beyond pixels using a learned similarity metric*. arXiv:1512.09300 [Cs, Stat]. Retrieved from <http://arxiv.org/abs/1512.09300>

Lewis, C., & Rieman, J. (1993). *Task-centered User Interface Design: A Practical Introduction*. Retrieved from <http://hcibib.org/tcuid/tcuid.pdf>

Mikolov, T., Yih, W., & Zweig, G. (2013). *Linguistic Regularities in Continuous Space Word Representations*. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 746–751). Atlanta, Georgia: Association for Computational Linguistics. Retrieved from <http://www.aclweb.org/anthology/N13-1090>

Nelson, H. G., & Stolterman, E. (2012). *The Design Way: Intentional Change in an Unpredictable World*. The MIT Press.

Newell, A., & Simon, H. A. (1972). *Human Problem Solving*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc.

Norman, D. A. (1986). *Cognitive engineering*. *User Centered System Design*, 31, 61.

Pieters, R., & Winiger, S. (2016, March 7). *Creative AI: On the Democratisation & Escalation of Creativity*. Retrieved 28 June 2018, from <https://medium.com/@creativeai/creativeai-9d4b2346faf3>

react: A declarative, efficient, and flexible JavaScript library for building user interfaces. (2018). JavaScript, Facebook. Retrieved from <https://github.com/facebook/react> (Original work published 2013)

BIBLIOGRAPHY

- Reed, S. E., Zhang, Y., Zhang, Y., & Lee, H. (2015). *Deep Visual Analogy-Making*. In *Advances in Neural Information Processing Systems* (pp. 1252–1260).
- Resnick, M., Myers, B., Nakakoji, K., Shneiderman, B., Pausch, R., Selker, T., & Eisenberg, M. (2005). *Design Principles for Tools to Support Creative Thinking*. Retrieved from <http://repository.cmu.edu/cgi/viewcontent.cgi?article=1822&context=isr>
- Rittel, H. W., & Webber, M. M. (1973). *Dilemmas in a General Theory of Planning*. *Policy Sciences*, 4(2), 155–169.
- Sage, A., Agustsson, E., Timofte, R., & Van Gool, L. (2017). *Logo Synthesis and Manipulation with Clustered Generative Adversarial Networks*. arXiv:1712.04407 [Cs, Stat]. Retrieved from <http://arxiv.org/abs/1712.04407>
- Salvador, T., Bell, G., & Anderson, K. (2010). *Design Ethnography*. *Design Management Journal (Former Series)*, 10(4), 35–41. <https://doi.org/10.1111/j.1948-7169.1999.tb00274.x>
- Sandberg, C. (2018). *react-color: :art: Color Pickers from Sketch, Photoshop, Chrome, Github, Twitter & more. JavaScript*. Retrieved from <https://github.com/casesandberg/react-color> (Original work published 2015)
- Schon, D. A. (1984). *The Reflective Practitioner: How Professionals Think In Action*. Basic Books.
- Shneiderman, B. (2000). *Creating Creativity: User Interfaces for Supporting Innovation*. *ACM Trans. Computer-Human Interaction*, 7(1), 114–138. <https://doi.org/10.1145/344949.345077>
- Shneiderman, B. (2007). *Creativity Support Tools: Accelerating Discovery and Innovation*. *Communications of the ACM*, 50(12), 20–32.
- Shneiderman, B. (2017). *Designing the user interface: strategies for effective human-computer interaction* / Shneiderman, Plaisant, Cohen, Jacobs, Elmqvist. (Sixth Edition.). Boston: Pearson.
- Simon, H. (1995). *Problem Forming, Problem Finding, and Problem Solving in Design*.

SPACESHEETS: DESIGN EXPERIMENTATION IN LATENT SPACE

Design and Systems; General Applications of Methodology, 3(828), 245–257.

smartgrid: neural network based grid layout. (2017). Python, Victoria University School of Design. Retrieved from <https://github.com/vusd/smartgrid> (Original work published 2017)

Stappers, P. J. (2007). *Doing Design as a Part of Doing Research.* In *Design Research Now* (pp. 81–91). Birkhäuser Basel. https://doi.org/10.1007/978-3-7643-8472-2_6

TensorFlow.js. (n.d.). Retrieved 6 July 2018, from <https://js.tensorflow.org>

Terry, M., & Mynatt, E. D. (2002). *Recognizing Creative Needs in User Interface Design.* In Proceedings of the 4th conference on Creativity & cognition (pp. 38–44). ACM. Retrieved from <http://dl.acm.org/citation.cfm?id=581718>

White, T. (2016). *Sampling Generative Networks.* arXiv:1609.04468 [Cs, Stat]. Retrieved from <http://arxiv.org/abs/1609.04468>

Woodbury, R. (2010). *Elements of parametric design* / Robert Woodbury ; with contributions by Onur Yüce Gün, Brady Peters and Mehdi (Roham) Sheikholeslami. London ; New York: Routledge.

Zsolnai-Fehér, K., Wonka, P., & Wimmer, M. (2018). *Gaussian Material Synthesis.* arXiv:1804.08369 [Cs, Stat]. <https://doi.org/10.1145/3197517.32013075>

