KEIO SFC

"Computational Resonance", measurement of Human-Computer Interaction in digital design tools that embed AI features

by

Joaquim Silvestre

A thesis submitted in partial fulfillment for the degree of Doctor of Philosophy

in the

Graduate School of Media and Governance
Environmental Design and Governance

7th February 2017

Declaration of Authorship

I, JOAQUIM SILVESTRE, declare that this thesis titled, 'Resonance frequencies of Computer-Designer oscillation in digital tool for architecture design' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: ________________________________

Date: ________________________________

“[...] design is about something that we had but do not have any more,[...]”

Kostas Terzidis

KEIO SFC

Abstract

Graduate School of Media and Governance
Environmental Design and Governance

Doctor of Philosophy

by Joaquim Silvestre

My original contribution to the knowledge of architectural design is the measurement method for the interaction with computers during the design process and the "Computational Resonance" concept. This one theorise the user's recognition of implicit channels for communication with a computer while the measurement method helps to analyse potential of digital tools. Designers that use the Algorithmic Design methods rely on computational procedures to develop a design solution. This measurement method can be used to evaluate their interaction when there is, in a part of their computational process, a Convolutional Neural Networks (ConvNet). Since the purpose of the Algorithmic Design method is to extend the use of digital tools by incorporating their computational nature in the design practice and thus, compute design solutions. This approach requires that design problems should be defined in a way that they can be solved through computation. Usually coding is used to manage this computational requirement. Through coding designers concretely describe to computers how an interaction should happen in order to include computations in their design process. In contrast to this, ConvNet uses indirect way to "encode" knowledge from examples. ConvNet is a field of research in Artificial Intelligence related to Machine Learning strategies that aims to 'teach' principles to a computer with example data. Teaching a computer is another way to programme, and thus to practise Algorithmic Design. The potential of ConvNet as a new kind of design tool to encode concepts of human perception difficult to describe and program is experimented. Conducted design experiments are based on new definition of interaction parameters and their corollary method of measurement. Through a study on digital tools and their relation with the designer during the design process, this thesis explain how this indirect way of encoding knowledge impact on the resonance of the Human-Computer Interaction.

Acknowledgements

Firstly, I would like to thank the Japanese Government for his generous scholarship that allowed me to fully work on this Thesis. My sincere gratitude goes to my advisor Prof. Ikeda, for all the challenges he offers to me. Without his motivation and support this Thesis couldn't be finished in time. I would like to thank Prof. Guéna for coming to Japan for helping me in my research. Beside my advisor, I'm thankful to my advisory group Wanglin Yan, Hiroto Kobayashi and Akira Wakita, their advice and insight were inspiring.

I'm grateful to all teachers, boss and acquaintance that through their connection brings me here in Japan for this PhD experience. Thanks to Frank Salama, to send me at the right place at the right time and Manuel Tardits who told me about the Monbukagakusho scholarship. To Valerie Châtelet and Kostas Terzidis and Shoei Matsukawa who, through a chain of acquaintance introduce me to my advisor. Particularly, I would like to thank the researcher who helped me during my work field : Anthony Pamart, Amine Bourki, Raghudeep Gadde, Sébastien Kuntz and finally Thomas Zedine and his co-worker at Petit construction. Those who participate to the experiment and native English speaker like Prof. Galloway who spent time to read and correct my writing.

Also I would like to thank my friends in Japan: Rémi, Patrice, Guillaume, Daniel, Nakamura, Haruna and Prof. Kensuke Hotta. Those in France who keeps helping me even when I'm far from them: Aurélie, Sébastien, Milovan and especially Guillaume Dufilho to listen my theories. I'm grateful to my family especially my mother for their unfailing support. I praise the patience and devotion my wife Saori who help me to keep a fighting spirit. Finally, I dedicate a thought to my grandfather, Louis Patois, and Uncle, Edmond Schmitt who couldn't be witnesses of this academic achievement during their lifetime.

Contents

List of Figures

Abbreviations

3D	**3** **D**imensions
AI	**A**rtificial **I**ntelligence
API	**A**pplication**P**rogramming **I**nterface
APS	**A**ctive**P**ixel **S**ensor
BIM	**B**uilding**I**nformation **M**odel
B-Rep	**B**oundary (-)**Rep**resentation
CAADRIA	**C**omputer-**A**ided **A**rchitectual **R**esearch **I**n **A**sia
CAD	**C**omputer-**A**ided **D**esign
CG	**C**omputer**G**raphics
CMOS	**C**omplementary**M**etal- **O**xide-**S**emiconductor
CNC	**C**omputerized**N**umerical **C**ontrol
CR	**C**omputational**R**esonance
CRF	**C**onditional **R**andom **F**ields
CSG	**C**onstructive **S**olid **G**eometry
CT	**C**omputational **T**hinking
DSLR	**D**igital **S**ingle **L**ens **R**eflex
Exif	**E**xchangeable **I**mage **F**ile format
GUI	**G**raphic**U**ser **I**nterface
I/O	**I**nput (/) **O**utput
IPD	**I**nter**P**upil **D**istance
IDE	**I**ntegrated**D**evelopment **E**nvironment
IFC	**I**ndustry **F**ondation **C**lass
OpenCV	**Open** **C**omputer **V**ision
OpenGL	**Open** **G**raphic **L**ibrary
OOP	**O**bject **O**riented **P**rogramming

OS	**O**perating **S**ystem
RGB	**R**ed **G**reen **B**lue
SfM	**S**tructure **f**rom **M**otion
SLAM	**S**imultaneous **L**ocalization **A**nd **M**apping
WYSIWYG	**W**hat **Y**ou **S**ee **I**s textbfWhat **Y**ou **G**et

Dedicated to my Uncle and Grand Father

Chapter 1

Introduction

1.1 Design process supported by tools

Architecture design is about solving creatively interrelated problems such as aesthetic, structure, layout, cultural symbolism and more. This design happens between idea in the mind and a representation on a support(Figure 1.3). The need for representation is the first introduction of tools in a process that could stay mysteriously enclose in the head like all other thought. Toolbox, tool chain, tool use, tool type can inform on the design process that occurs behind the skull. Through the study of tools, the knowledge about the design process progress.

1.2 Introduction

Architects are doomed to try to solve as best as they can wicked problems [1] of creating an artificial environment that doesn't negate the natural environment that receive it. They have to create a link between the natural and the artificial despite their antagonism. Those wicked problems are interrelated with tamed sub-problems. To solve the tamed ones they can use established science, while wicked ones have to be undertaken with a failure risk. Nonetheless, concerning the tamed ones, they have to think on their feet in order to keep up to date their knowledge about the latest optimal resolution procedure. For the general wicked problems, they can try to reconfigure the problem statement and organise relations between tamed and wicked issue in order to find a configuration that can be solved in a once. To help in this challenge, they use tools of indirect design to create the representation of the solution separated from its production. Nonetheless, representation is a production work itself and demands direct tools [2]. "Direct tools" are a kind of tools that help in the decision process. It manage the resolution of those tamed problems automatically and on demand. They directly propose the decision to take while i"ndirect tools" are used to create either a direct tool or a document that will serve as a ground for a reflection about the solution. For those tasks, solving and representing, designers entered [3] in the digital age and use the polymorphic tool known as a computer. Computer itself is not a tool, it's a platform for computation. And through software the hardware computation is controlled. All the software layer is an aggregation of multiple and various programs that interact together. When designers stand in front of their favourite software's GUI they don't see legion of micro-tools interacting together to create their user experience. They see and they call a "tool" the software they have in front of their eyes. In fact it's often a collection of tools. Some are creating the visual experience, some manage mundane low-level task in background and some effectively do the work expected. This situation of design creates a feedback loop of interaction. The designer sends input, see output takes decision and restart the loop(Figure 1.12). From each output feedback, there is a readjustment of the intention. This loop is true with all feedback from digital tools, even when they are not used in the design process, after a conceptualisation phase designers create a visualisation. From this visualisation, new idea emerges and a new conceptualisation phase starts. This cybernetic approach of the design takes another turn with the computational tool. Since programming tool is a tool that makes tools, the designer can create their tools and build-up their own design environment.

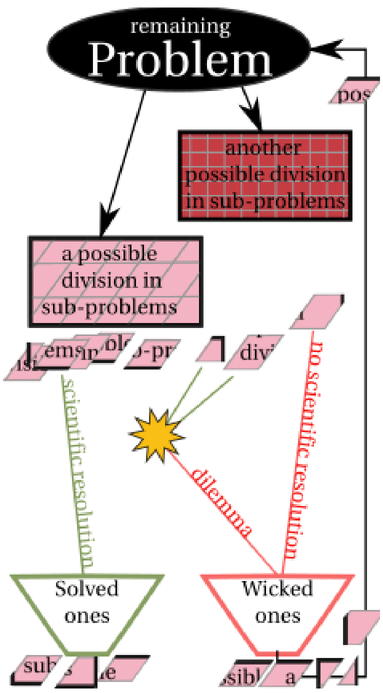


FIGURE 1.1: Wickedness of those problems can be in their decomposition, in the conflict between solutions of sub-problems or just because one sub problem is inherently a wicked one

The stakes of the environment are clearly emphasised in historical materialism theory for civilisation process. At the individual level, theory of the extended mind [4] advocate for situating memory in direct environment. Our surroundings are a part of our mindset because it offers information within sight. Therefore by changing environment, a part of the mind is changing and this view is even more strongly defended by Lambros Malafouris [5]. Even if those changes can't be deep enough to change the personality as Arnold H. Buss[6] understand it, they change the consciousness of abilities on the place and time of action. The computer offer a visual and operative environment that can be a part of such mind extension. This idea was formulated for all kinds of production tool by Flusser :

> *Tools change our behaviour, and thus our thinking, feeling and wanting. They are experience models. [7]*

This idea of empowerment of the designer face to its digital tool raise with the advent of algorithmic design practice and theory.

1.3 Research Background

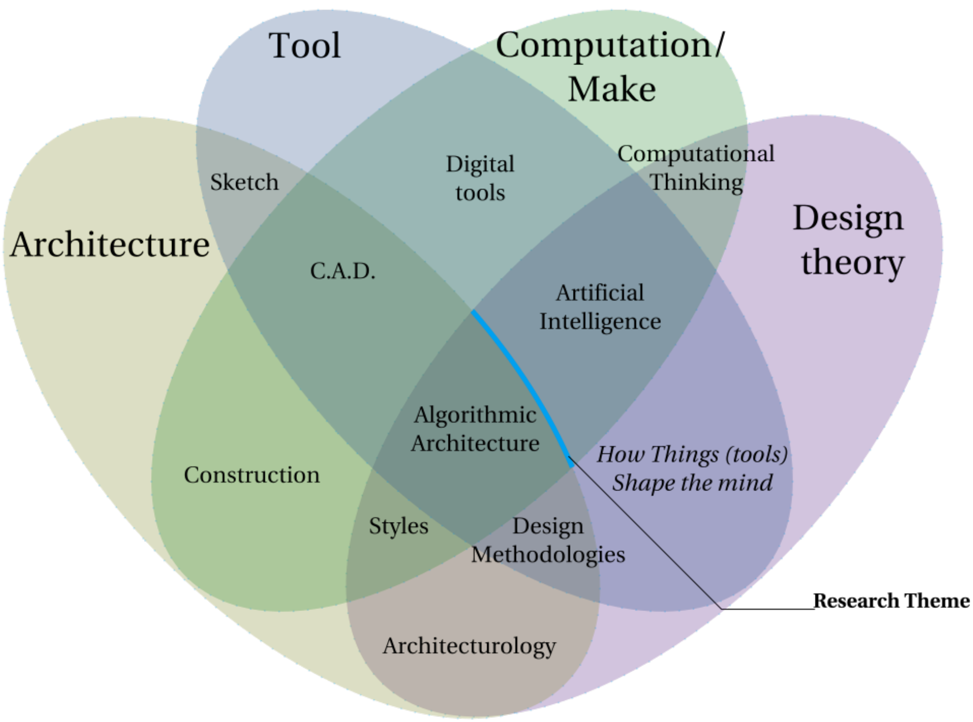


FIGURE 1.2: This research look on the interface between algorithmic design method and A.I. research

The background is extensive (Figure 1.2) since it connects topics of Artificial Intelligence, Computer Science, computer vision, Computer Graphics, Design Process theory and Algorithmic Architecture. In each topic a specific part is useful to understand the result of this research. To condense the description of the research background, four appendices will cover in depth the topic of Design software culture, Convolutional Neural network in Artificial Intelligence, 3D capture technologies in Computer Science and Architecturology in study of Design process. The reader can refer to these appendix to get more detailed information and explanation on the topic. Those descriptions are built upon experience, course and readings. As much as possible sources have been cited in these appendices. Nonetheless, they don't belong to the thesis as reference of other works but more as a help to get an easy access to common knowledge of each field. The Appendix A centralise all the tools mentioned in this thesis. For conveniences they are organised by categories of software and in each software tools are presented. Stand alone tools are gathered by the knowledge field. Other appendices are a summary of useful knowledge to read this thesis. According to its profile, reader will look for explanation in the appropriate appendix.

Appendix B	ConvNet in A.I.	principles and componement of ConvNet detailled here.
Appendix C	3D capture technologies in Computer Science	format of 3D model, Computer vision principles and Photogrammetry working principles.
Appendix D	summary of Architecturology principles	translation and explanation of the Architecturology vocabulary

The main and central background of this research is Design process study field. All the other topics such as Machine learning, 3D modelling or programming are presented as applied in this field. Those topics are introduced in the thesis because they are the domain knowledge from where key component of the tools have been built. The tools that will be mentioned in this thesis are all kinds of digital tools that can be used for designing. The thesis focus especially on tools that open on creative outcomes through computational process. We will call the phenomenon of creative outcomes with these tools : Computational Creativity[8]. A second part of the research background will explain the background of those tools, the reference literature of the domain knowledge they are made of. In this part a conceptual framework based on "Computational Thinking" [9] will be presented. It will offer another definition of digital tools and an evaluation system of their use. This framework will serve to frame the experiments conducted in the chapter 3.

1.4 Design as a Science

Architecture design may not be gave up to the fallacy of artistic licence. The workable separation of Art, Science, Engineering and Design that have their own sociological process of validation by their peer prevent for cross-disciplinary or even anti-disciplinary approach [10] that are the only one able to overcome the problem without creating other indirectly. Scientific solving of design problems have been proposed by Christopher Alexander[11] [12] in the reduced field of architecture. The implementation of such encompassing approach is itself a challenging wicked problem since it needs to take in consideration how problem are effectively solved [13] and not only how they should be. Indeed, the scientific solving is not a theoretical solving, it needs to be integrated in the concrete social reality. Face to such complexity, the discovery of a holistic solution may require the assistance of an Intelligence of another nature which can see what is beyond human horizon. First attempts and hope and AI were limiting it to the problem involving intelligence. But the understanding of intelligence evolve and open the hope to even have creative AI [14] and thus, creative solving of design problems.

1.4.1 Design as a Wicked problem management

What is solving a problem when a recipe or an algorithm exists to solve it? Indeed, such problem should be renamed into solvable problems in order to underline that the real nature of this problem is just about finding the existing method to solve it. Those existing methods or algorithms, if they rely on scientific principle and not on dogma or policy, should be provable. The archetype of the perfect problem resolution is the mathematical problem. Every element used to solve is proven token and the solution will always come out if the suitable method is selected.

We may hope that one day science will cover all problems that we can encounter and provide us methods to solve them all. Meanwhile humankind has to tackle down problems that are not scientifically solvable. Real problems that need a science of solving the temporary unsolvable problems. This science, or practice, is Design. According to the discourse of method

> *"Divide each difficulty into as many parts as is feasible and necessary to resolve it."[15]*

If the designer follows this way, his first problem is to divide the problem. And this question is open to debate since there is no such science. Indeed division implies that the subject is clearly defined and a characteristic of wicked problems of not being correctly defined. At least, the designer can recognise that he is facing a wicked problem when the problem has one of 10 characteristics of wicked problems. At this point, he knows, if his bounded rationality takes account of the notion of wicked problem, that he will not find a solution but he will have to decide a dilemma in informed posture and do as best he can. Melvin M.Webber described the wicked problem in contrast to the tamed ones. Those tamed are "the scientifically solvable one" mentioned later. By supposing that the problem can be divided and organised into tamed sub-problem and wicked ones. Even the interaction of tamed sub-problems can become a wicked one. Finally, design solve by merging sciences when possible or use heuristics that have been proven by the experience of the field being worthy.

If we look at the etymological definition of design provided by Terzidis Kostas[16], the Latin roots point toward that the word is about "derivation" of something and the Greek roots indicate incompleteness but also "the strive to capture the elusive". Indeed, further back in the origin of the word :

> *as such, it is about an event that did occur at an unspecified time in the past, hence it could have happened any time between a fraction of a second and years ago. So, according to the Greeks, design is linked indirectly to a loss of possession and a search into an oblivious state of memory. This linguistic connection reveals an antithetical attitude towards design that, in the Western culture at least, is about stepping into the future, a search for new entities, processes, and forms, frequently expressed by the terms novelty or innovation.*

Regarding such definition and by according some credits to the wise of ancient Greek, we may extrapolate that the solution of the problem in design is in the past, the experience of the designer and it can be extended to all humanity through the sum of scientific knowledge. This history of solutions include all the possible science and the designer, rightly or wrongly, take decision in his bounded rationality. The bounded reality of his scientific and historical knowledge that he tries to retrieve but had lost–that is not present here and now when he faces the design problem.

1.4.2 Design Process Study Field

One science may help the designer to know about his choice : the science of design. It indeed study the process of design and attempt to develop method of design that can be seen as theories of the good design that are, like scientific theories, pending for observation that comes to point out anomaly of that good, or better, method of design. That's why this field is also called: Design Theories and Methods.

Firstly, beside the analysis of problems in design by Rittle another design theory that form the background of the thesis are the research of Herbert Simon[17]. The elaboration of a design solution is studied as a theory of decision-making in his seminal book: the science of the artificial[17]. Simon propose to emphasis on the limits of knowledge of alternatives and their consequence in decision making in his bounded rationality theory. Even if the right decision can't be taken, a science can study how decision are taken and look retroactively how the best have been taken. Simon proposed a model of decision taking that was oriented for an application in the economic field. Nonetheless, the key idea of this theory are valuable for research about design.

This notion of being bounded to a portion of our own rationality found its translation in the term "experience" in Boundon's work. Boudon, like Simon, didn't propose a method for design in architecture. He proposes a vocabulary to define the operation, moment, concept of designing in architecture. The specialisation of this vocabulary for the architecture field allows term more precise to describe the aspects of the design in architecture. Indeed, the decision taken by architect are made in a specific context: creating something that have a material reality. Unlike a decision taken by a product designer, the architecture designer production takes place on earth at the human scale. This scale question is the key feature of architecture. Indeed, manufactured product are at human scale since they are used by humans. Nonetheless, the uniqueness of the human scale in architecture is that scale allows the individuals to go in it and becomes a complete surrounding of his senses, an enveloping experience.

The epistemological goal of Boudon is to define the architecture as a scientific field. According to its book "Sur l'espace architectural[18]", the specificity of the architecture is the science of designed, and thus artificial, form. In this thesis, the author will use the term "conception operation"(Figure 1.4) and "measure attribution". Those two terms are defined in the Appendix D but to give a brief explanation, a conception operation is a mental operation that is applied on a model. This operation belongs to one of the 16 elementary scale or one of the 5 meta-scale[1]. An operation is an attribution of measure, a kind of sub-problem that is belonging to a scale category. This sub-problem resolution can be compared to a bounded rationality case since similarly a comparison to "reference" occurs in the Architecturology's description system. Those references used as point of comparison match with the "intelligence gathering" stage of Simon's decision making model. Architect designer that try to identify alternative through references encounter the same issues pointed out by Simon: they can't be all listed and the consequences of each alternative can't be all foreseen.

[1]The Architecturology is open to new scales. Despite the original scales may cover all type of conception operation, Boudon is open to add scale if an operation can't fit in any of the existing ones

Thanks to Architecturology vocabulary, the use of digital tools can be located in the process as happening during the conception operation. One or several tool can be use to achieve a design operation. Tools can be use to solve multiple scale measure attributions in once. In this case, this is a conception operation that occurs in a meta-scale. We can delineate two main use of tools:

- They can be use to the representation of the result of the operation. In this case the design operation occurs in the *design space*
- They can be use to decide the measure attribution. The tool contain some quality to decide the pertinence automatically. It can be a proven algorithm or a well-known heuristics encoded by programmers in the digital tool.

Those two case can be overlapping since the representation help to foresee some consequences of the measure attribution. By this way it is an "help" to measure attribution. On the other side, the digital tool never "decide" the measure attribution. The designer decide to rely on the digital tool. And when the output came to sight he takes the decision to keep it or leave it. So the decision is still implicitly up to the designer and both category are about helping for decision. Nonetheless, we can suppose that the use of the tool can be done with one main intention initially. Others description methods show other advantage and help to focus the analysis of the design process on different aspect of the phenomenon. Globally the vocabulary of those system are slightly different but they present the same type of relation between the term. This description system is chosen amongst other because it is specialised to architecture.

1.5 Power of Creativity in Design

This problem solving vision of design tend to mask the role of creativity. This term, as much as the "intelligence" term, is too ill defined. Indeed, due to its overuse in the common language, the creativity can relate as much to a lack of culture of the observer than a biased introspection of his own taste. Concerning the creator side, it could be an excuse for certain licence or lack of the ability to track his own inspirational source. Fortunately, this seemingly vague concept has be theorised by [14]. She proposes to model creative process as concept space that is explored. This space has boundaries that characterise edge of understanding. Beyond this edge a there are unintelligible chaotic expressions that are out of creativity. The "center" of this space is where most common ideas rely. They are not surprising and don't require creative effort to be discovered. Creative value of idea increase close to the concept space frontiers. This is on the edge of understandability that valuable creative idea need an effort of exploration. When rules of the concept space can be formalized, there an opportunity to automatise the exploration through an algorithm. This one of the approach of digital creativity practice. Those rules of space can be source of creative thinking too. The creativity produced through this transformation of rules is called *transformative creativity* that is distinguishable from *explorative creativity*. Wiggin [19] propose a formalisation of this creative system description that takes in consideration the language $\mathcal{L}$ used to create the rules $\mathcal{R}$ that set this conceptual space. The navigation in the sea of possibilities in the conceptual space follow a traversal function $\mathcal{T}$. Each items encounter in this space are considered through a evaluation function $\mathcal{E}$. Transformational creativity may be achieved by modifying one of the function$\mathcal{E}$, , or rules$\mathcal{R}$ or it means of expression $\mathcal{L}$. This definition is loose and can be used to describe the design exploration process. Design as a problem solving can be seen as a sort of creative transformation with tighter evaluation function $\mathcal{E}$ of the discovered item: they have to be a solution of the problematic situation.

1.5.1 Intelligence or Creativity

The vulgate use of creativity and intelligence terms employ them to define human qualities that are distinguishable. Solely intelligent peoples and solely creative peoples exist, in a more restricted proportion, some peoples seems to posses the both qualities. Regarding solving as a proof of intelligence, traversing the conceptual space to find a solution is quite similar to traversing it to find novelty. If a solution exist in the center of the conceptual space, we couldn't say that the one who found it is intelligent. As Alison Pease conclude in her paper [20], there is a need of a clearer subdivision of the quality we want to emulate digitally:

> *The question which faces us now is whether we should see creativity as analogous to intelligence or as a particular aspect of it. In either case we need to subdivide it further, but what these sub-aspects mean in terms of overall creativity would differ. If it is seen as analogous to intelligence then evaluation of creativity may turn out to be based on our best yet test for intelligence - the Turing Test.*

With a doubt concerning the supremacy of the Turing test [21], the importance to hone the subdivision of the creative intelligence is shared by the author. The clear and precise subdivision help to select more precise evaluation of the success or failure in emulating the "human" quality with a machine. According to Alison Pease paper, the "creative intelligence" researched in this thesis get is value through the evaluation of output item quality relative to an aim. An aim of creativity and productivity that let room for the randomness in the method of generation if it can generate novelty relative to complexity. This evaluation can be experienced through a conceptual shift [22] or a situation of computational resonance.

1.6 Role of tools in the Design process

The thinker of the design studies had long time ago envisioned what could be the link between the digital tool and the designer. 1962 in England, at the "The Conference on Systematic and Intuitive Methods in Engineering, Industrial Design, Architecture and Communications." Jhon Chris Jones stated that:

> *"The ideal picture of a man-machine symbosis is ... one in which machine and human intelligences are linked into a quickly responding network that permits rapid access to all published information ... The nett (sic) effect is expected to be one of mutual stimulation in which open minded people and progammes nudge each other into unpredictable, novel but realistic explorations ..."[23]*

Indeed, the idea of associating power of digital tool to counterbalance designer's weakness face to the listing of alternative or the problem generated by the interaction of sub-solutions. Beside the calculation, an asset of the computer is the storage of information. But storage without a powerful means of research is not very valuable, hopefully, computation allows through algorithm to retrieve the information with keyword, category and a multitude of means. But even if good classification system help to reduce the number of keyword necessary to retrieve the good reference for a specific problem, it still need to be remembered by the designer.

Human memory is the key to access reference, and theory of mind propose different view of its functioning. How and where are stocked information is a large question in cognitive science and philosophy. A philosophical background reference that frame this research is the theory of the *Extended mind*[4]. This theory proposes a non-Cartesian vision of the cognitive science. Cartesian cognitive science stated that cognition is manipulation of information about the world through mental representations that are corresponding to a physical structure in the brain. According to this point of view, all things in thought pass through a mental representation construction phase. The non-Cartesian alternative propose to acknowledge *external processes* as literally part of some cognitive processes. To support this thesis, Rowlands, define the external processes, and processes in general, as exploiting information bear by structures. The external processes exploit, indeed, external structures. External is here relating to out of the skull but connected to it through the senses provided by the embodiment of the subject. Even if part of the cognition is outside the skull, the author keeps the ordinary assumption that cognition is done by a subject. Therefore, the external processes are only a part of the cognitive processes and there is still an irreducible *internal-neural* embodied part that is part of all mental processes. *External processes* are sub-processes integrated to personal level process when they make contribution to it. And the *epistemic authority* over them allow to recognize them as personal and contributing processes of extended process. The *epistemic authority* is the acquaintance to the relevant working of the bear information structures. This acquaintance pass through vehicle of cognition that are illustrated in the book with example such as the feedback of cane of blind person, the feedback of saccadic eye movements or the feedback of sensorimotor contingencies. By considering that mind frontier are beyond the skull space. And in order to extend mind to the surroundings, the mind have to be embodied in fashion inspired by *the ecological theory of visual perception*[24] and use the manipulation of the optic array to provide the necessary feedback to recognize the *epistemic authority* over processes. We will see through further developments that those external processes are related to use of digital design tools as vehicle of cognition when conception operation are processed. Then, with use of cybernetic theory a resolution of the problem of epistemic authority as a non-permissive[4] criterion for recognition of ownership of process will be proposed. Beside the notion of process, the static aspect of memory can be explored with Rowland's concepts. Indeed, surrounding provide a context of thought where the consciousness is picking information here and in brain memory in order to create new thought. Memory have been related to surrounding space since palace of memory have been used as mnemonic technique in order to enable orator to remember long speech.

If the *Extended mind* theory is invoked here, it's firstly as a new framework to understand the mind and secondly because it is part of the theories used by Lambros Malafouris in his "How Things Shape the Mind" and we will see that "things" can indeed be tools. Malafouris defend a theory of *Material Engagement* that ground reflections on archaeological and anthropological findings in order to explain the problem of agency and creativity. This work of cognitive archaeology gives a more specialised view of an extended mind approach of the tools. Those tools can be for production or ideas but we will see in the Chapter 2 that delineation doesn't have to exist. Tools shapes the mind and they are "made for" ideas. In fact we don't know if they are "made for" a reason but their practice provide ideas and those idea want to be controlled in order to push them to the extreme boundaries of our rationality during decision taking.

If "Things shape the mind", tools may somewhat shape idea. Christian Gänshirt warn about tools that can "falsify" our ideas[25]. Despite they are necessary to remotely convey ideas that cannot be directly transferred from a mind to another. Or in a "extended mind" vision, disembodied in order to be re-embodied by passing through physical structure. As his book present different tools with their own rules that misunderstood can make them independent from the designer. This vision designer-centric try to aware designers to the tools flaws and strong point in order to "control" how the tools shape the mind. The intent is to avoid that tool control the mind in a undesired way. The flaws are seen as the part of the tool that "trick" the designer. This research focus on creating the conditions for emergences of idea without following a vision of a supremacy of the original idea of the designer that may leak out from Gänshirt's vision of the position of tools in design. Nonetheless, he gives a fresh and insightful analysis of the influence of tools on the mind. He proposes a division of tool in two categories : Visual tools and Verbal tools. Visual tools are those that involve gesture while verbal are based on the word. Those "physical" tool that shape the mind are mainly in the visual side. Nonetheless, it is worthily to note that computer belongs to the both categories.

1.6.1 Mind theory for tools with mind

As demonstrated here, theory of cognition help to understand the effect of tools on design. Research on digital creativity search for applying to the design of digital design tool some insights of the Mind theory. Nicholas Davis[26] propose a model of the creative process, the Enactive Model of Creativity (EMC), that support a similar view to Mark Rowland's theory of the extended mind. Base on this model, he proposes to reframe creativity process understanding for the design of future tools. By adopting this model, the problem of Digital Creativity (DC) and AI encountered during the development of such tool can be overcome. A key component of this model is the "perceptual logic" that filter the perception of the world according to the creator's mental model and the previous feedback from this world context where action is occurring. The "perceptual logic" is what the digital tool have to affect in order to improve design and creative opportunities. This perceptual logic present different layers that can't be affected in the same way by existing algorithm and HCI. While local layer can focus on creator contribution, global layer is harder to manage if the scale of the globalness is extended. Another aspect of tools based on this model could be to clamp and unclamp the perception during the process. Nicholas Davis envision a whole set of features of the Enacted creative mind that can be supported by digital creative tool. This vision of designing tool that support actively creativity from the origin of their design process present positive issue that had been well evaluated by other thinkers[27].

1.6.2 Digital tool for design

The evolution of background from things to tools, influencing mind therefore idea, bring to the last step of this progressive focus : the digital tool. As seen through Gänshirt's categorisation, digital tools are as much visual tool than verbal tool. The polymorphism of computer is such that it can be both. But to progress in the detail of the concepts of digital design tools, the nature of computer should be explained here. The computer itself is a tool for computation. But with his programmable feature, during interaction with user, it can triggers scripted control loop feedback. The computer as programmable tool show is ductile aspect towards organisation of computation. Nonetheless, as programmed tool, it is a efficient mind changer. Despite that originally it is a programmable tool, it is never blank from any program so its malleability is double-edged if the influence of tools is seen as harmful for designer's original idea. By looking it as a programmed tool, computational tool cover still a too large meaning that even if all programmed tool use computation, all the tool are not about a conscious control of the computation by the designer. For this reason, they are group under the term of digital tools that emphasis through the word "digital" on their encoded aspect more than putting the computation at the core of their interaction principle. Most of the digital tools try to hide the computation to the user with interface that help to shift the stream of focus on the input necessary for the operation. A large part of these digital tools are 3D modelling representation tools.

A thesis written by Aurélie de Boissieu[28] characterise the cognitive operation of conception used in parametric modelling tools. And through those description, a mapping of the new abilities encoded in the extended mind of the designer can be figured. This thesis present an overview of what kind of tools are in the designer digital toolbox out of the shelves.

1.6.3 Digital tool for creativity

The Computational creativity field presents categories and classifications of digital tools for creative practice. Proposed by different authors, they don't belong to a comprehensive and unified taxonomy. Nonetheless, those various categories system can help to situate existing digital tools and orient the figure out how new ones could behave. Ultimately, those different system show that the creative orientation undermine the importance of some features for the design process. Indeed, tools such as the "*Drawing Apprentice*" or the "*Wekinator*" show powerful means of exploration $\mathcal{T}$ of concept space while letting the evaluation $\mathcal{E}$ up to a logical perception up to the creator's sensibility. Such tool are creative experience oriented while design tool discussed in this thesis point toward a methodological and ethical paradigm: the computational thinking. Nonetheless, categories of CC are powerful to describe digital tool and I will present here some categories that will help to describe tools presented in this thesis.

First of all, it important to remind that all tools are free of any meaning. User give purpose to them and elects them as tool for productivity or creativity. An artificial distinction that [29] depend of the task. From a design document production perspective, tools can be both productive and creative enhancer and in some way we can argue that everything is about productivity. If the tool make the designer hyper productive the traversal of the concept space is faster thus it is a creative support tool. In a analogue approach, a creative support tool is a productive support tool if it bring directly to the creative output without trying large amount of unsatisfactory item. This vision seems to obliterate the effort aspect of the cognitive work of design. Indeed, a creative support tool can bring to think an idea but attaining the idea in mind doesn't imply the skill to produce efficiently the document for is communication. The communication aspect often problematic in the Creativity Support Index evaluation system illustrate that tension between a pure concern of creative experience and the need to meet some standard for communication and conviction necessary for collaboration. That vision of a productivity that possibly overcome creativity requirement is contested by the analogy proposed by Kumiyo Nakaoji[30]. He presents CST as belonging to three categories: dumbells, running shoes and skis. Dumbells train the user, after a certain period, the user may acquire creative skills and emancipate himself from the Dumbells CST. The running shoes allow to do something the creator can already without the CST. With the CST he or she can do it faster. In some extent, this quantitative improvement lead to some qualitative enhancement. The ski CST allow a completely new creative experience that is impossible without the tool. In this case we clearly see that a productivity support tool that offer a "skiing" experience is de facto a CST too.

As Erin Cherry and Celine Latulipe point out, those CST can belong to a larger set of CST that can be called Creative Support Environment CSE. Such environment is modifying the perceptual logic by offering a set of affordance within the enaction. Since CSEs may requiere dedicated hardware or instrumented space, they are ski CST type too. Another classification of creative tools is based on the modality of the relation with the computer. Todd Lubart[31] propose to classify the Human-Computer modalities of interaction presented in a special issue about "Computer support for creativity". He proposes relations that personify the computer : computer as nanny, as pen pal, as coach, as colleague. For instance, computer as coach is very similar to the category of "dumbells tools", nonetheless, the creative tools that transform the computer as nanny for creative process is not exactly like running shoes category but clearly different from the ski and dumbells categories.

Another classification based on the abilities of the digital tools present three large categories : Creative Support tools (that for sake of clarity we will call Basic CST), Generative Computational Creativity (GCC) and Computer colleagues. Computer colleagues is the blend of the two previous categories. It seems to be the touchstone of creative tool designer. Basic CST regroup what officially is intended to be creativity support tool: Photoshop, Illustrator, GIMP. GCC are more autonomous but seldom present on software market. It is maybe because they invert the usual Human-Computer relation.

1.6.3.1 The role of tools in the conception visualisation loop

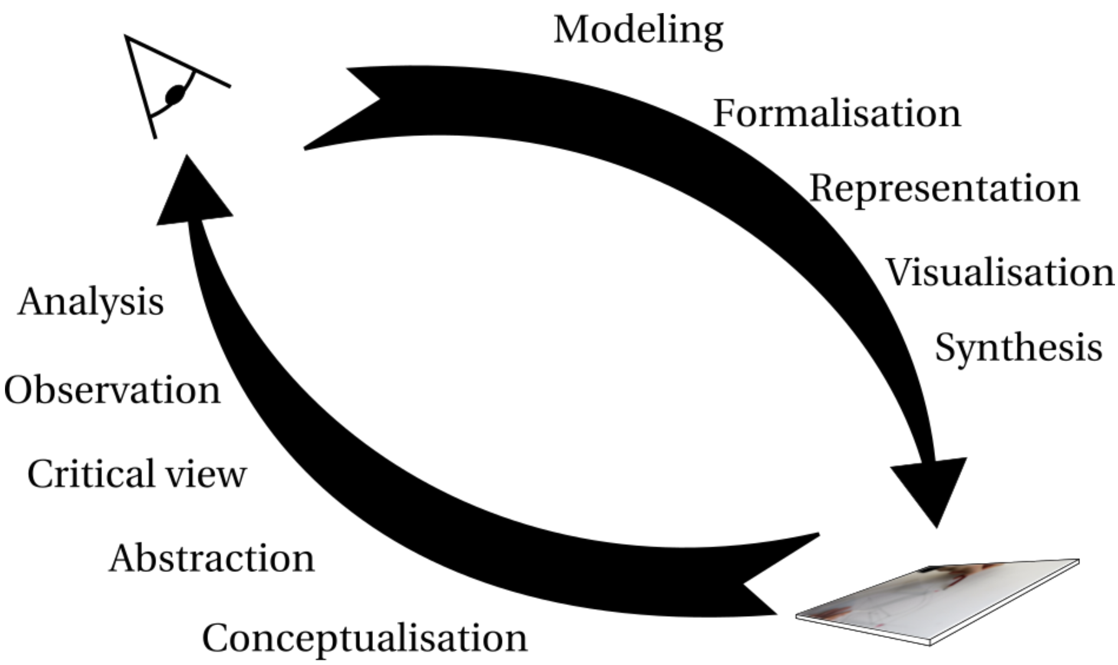


FIGURE 1.3: Various terms are used to describe the two phases of this process.

Like in Figure 1.3, the design process can be described as a cycle of conceptualisation and visualisation [32]. Visualisation helps through the materialisation of the concept to refine the detail of its effective realisation[33]. The process of producing the visualisation induce question in the mind of the designer [34]. Those questions are hardly foreseeable without a medium that lay down the main idea of the designer while triggering interrogation about the flaws of this idea. The irrelevant points of the current design proposal are extracted through a critical eye on the document produced to visualise the designer's idea.

Before being a visualisation, the idea is part of the thinking subject. Visualisation objectify the idea and thus put it at a distance from the designer. From a distant point of view he or she can develop multiple perspective with a constructive perception [35] attitude. This critical thinking provides the seeds for a subsequent conceptualisation of the design problem. This visualisation pass through a formalisation that " enable conversations with oneself as well as with others" [36].
Designers may rely on digital tools to outsource or simplify part of the visualisation production task. Those tools are based on algorithms that preform very precise task without any autonomy. Everything needs to be specified and unspecified parameters are set by default. There is not enough "intelligence" in the digital tool to sense and adapt itself to the context. At best, if well designed, it can cope with the management of errors related to the computational context but it can't autonomously guess designer's intention concerning the design context. Architects and designers that practise algorithmic design [16] tend to have constructed perception of their tools and their contexts is already encompassing the computational one.
When they design, computation is part of their conceptualisation of the process itself thus the visualisation production is managed in such fashion that the design happens through this computation. The CR that can happen through such process can be envisioned as kind of premeditated CR. In the case of this paper, we will see that more "spontaneous" case of CR can happen when AI components are involved.
Progress in the understanding of the CR can lead to the improvement of the visualisation phase through the significant diversification of its production process. Through their autonomy and relative independence, visualisation produced by AI based tools present serendipity effect through mistakes and unforeseeable misinterpretations due to their working context. In the case of the ConvNet based tools, their context overlap the sensitive world context in its visual aspect and the computational one. This overlapping of context is the significance of the diversification provided by such tools.

1.6.4 Computational thinking to articulate tamed problems

The science of design exposed until here where about model trying to describe design in a general way without presupposing any design methodology. Those models try to relate the "natural" design without developing the case where tools are used. Then other theories came to focus on the case of tools in general, then for design and finally digital ones. As Gänshirt warn about the tool but it doesn't provide a methodology for tool use in design or more precisely, how to use which tool and when during the design process. His book was about a general presentation and the number of possible tools for design don't allow the development of a comprehensive method that encompass all possible tools. Even the recent thesis on parametric tool, a narrow set of tools" didn't develop explicit methodology for design. Great designer, such as Bruneleschi or Leonard de Vinci had probably invented their own methodology that fit to their character and talent. Nonetheless, the exponential accumulation of digital tools in the computer may leave the designer lost face to the functioning of the tool itself. Explaining the tool and explaining a methodology of design that employ the tool are two things different. Even if software marketer want to promote the idea that the design methodology is up to the user, the reality is far different. The tool impose a process of use, a logic, derived from the computer logic, that need to be assimilated by the user in order to use it.

Woodburry, in his book Elements of Parametric Design[37] propose a methodology at the half way between a tool explanation and a methodology of design. It's a methodology for designing parametric model in a specific software. So it explain the tool, a methodology that empower the user in the creation but let the user free to find a way to merge this method with his own design method. Computational thinking illustrated [2] is not a academic publication but teaching material for popularising this problem approach and solving methodology. Nonetheless, it has been featured in the "The CS Principles Project"[38]. It propose 4 categories with examples of processes related to this methodology :

- Decomposition
- Pattern Recognition
- Pattern Generalisation Abstraction
- Algorithm Design

1.6.5 Programming as design tool

Without claiming belonging to the computational thinking approach of problems, some designer advocate for the use of programming in the tool set of the designer. For Gänshirt this type of tool is typically belonging to the verbal tool of the computer, but since visual output can be produced on the basis of this verbal tool, the design situation that use code and graphic output are in-between design tools. Despite the use of code give a

[2] http://www.ctillustrated.com

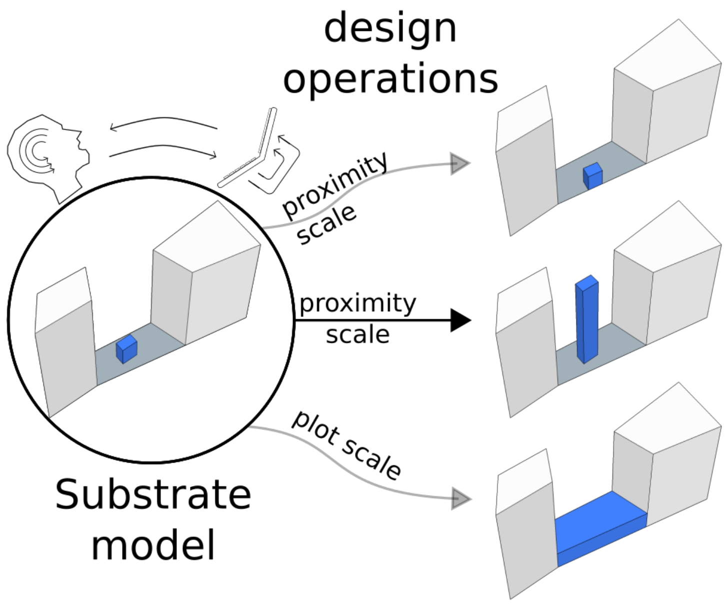


FIGURE 1.4: Control loop feed back happens in the conception operation phase

more direct access to the computation, it doesn't imply a "computational thinking". It is indeed thinking with a computer but not thinking with strategies of computer scientist. As said before, computational thinking does not imply the use of a computer. A problem can be solved with computational methodology without computer involved in the process. In the same way, computational thinking methodology neither imply the use of code. Parametric design is already a form of computational thinking since it model is based on a strategy of *pattern generalisation*. Nonetheless, programming as a design tool is still a slightly different case of digital tools compare to the other digital tools. Indeed, CAD are a metaphor of design board while parametric modelling system are already close to computational thinking but does not present the singular aspect of mixing verbal end visual characteristic. This feature, in a material engagement framework, has greater impact that tools that "stay" in their category. They shape the mind but they can shape tools with the intent to shape the mind afterwards. This recursive and remote design is a figure of meta-design or design of the design process. Code in design is often used as this way. As we will see later with example, designer that employs those technique doesn't aim to only compute the solution and get rid of it. They want to install themselves in the way in peculiar way. With or without the intent to shape indirectly their mind later, they create intermediary tools between them, the computer, the model and the design process itself.

1.7 Computation in design method

We've seen that neither computer or code are not unquestionable sign of computational thinking method of problem solving. Above that, we've seen that usually, designer may use computation, and computational method but in some case they "organize" a design environment based on computational thinking but keep themselves in the control loop feedback. Examples of case where the computation was organize in such a way that the design solution is computed at a certain step of the design process and the designer intervention is just in a validation position. Two iconic examples of the use of

FIGURE 1.5: An example of computational design, the olympic pool for Beijing 2008 games. Designed by PTW Architects, CSCEC,CCDI and Arup

computation in design method can illustrate case of organization of computation rather than having a material engagement with the code. The stadium (Figure 1.6) and the pool(Figure 1.5) of the Beijing 2008 Olympic games are both partly product of the computation. Partly, cause initial idea is a non-computational metaphor, it's a "pure" designer idea that is adequately transposed into a representation for fabrication. Here, the computation had computed the details of the plan drawing. The concept it self is not computational and the computation is peripheral in this methodology. Indeed, the computational thinking here is used to solve solely not to design. The nuance may be due to the absence of computation in the loop that created the main concept that leaded the project. The nest bird and the voronoi where not created in a meta-designed computational context. In some way, since the solution is not completely produced by scientific means and the problem is not completely defined as one unity of necessary interdependent problem, the computation can be relaunch and a loop of interaction exist. Above that, despite the condition of the interaction, the planing of a chain of digital tool that produce the solution is already a meta-design of computation for design.

FIGURE 1.6: An example of computational design, the olympic stadium for Beijing 2008 games. Designed by Herzog & de Meuron, Arup Sport, China Architectural Design and Research Group, Ai Weiwei

1.7.1 Computational design by permutation

When code is used, the computer becomes a machine to create tools and thus, the designer design is design environment. He becomes digital design process designer. Aware of this singular situation, architects with programming background have made experiment on the practice of personally designed design environment or process. Those experiments are not so meant to answer to a specific brief and they are more free problem setting that are "solved" with self-made computational design tools in a computational thinking approach. The first examples to illustrates this peculiar practice is work of researcher/designer Kostas with Permutation architecture are seen by the author as sign of "design requirement"[39] that doesn't focus on the output but the how to produce by computation a solution and its representation. The concept of this process is to try all the possible solution of a reduced problem and eliminate all the solutions that are not fulfilling requirement. The core requirement is to be viable solution. On this base, the problem is defined in a way such as computational method can solve it. In this case, it limit is design space to square shape bathroom of 4 square meters. That minimal bathroom can contain 4 components that can be oriented in 4 way. Same as method to create I.A. that play chess, the system is based on the idea of testing all possible solution. Such method encounter always the same challenge, the computer memory size. The architecture problem, according to drastic restriction of the initial parameters of all possible bathroom, find a viable design for a minimal bathroom, becomes a problem of computer memory management. This memory restriction is a sub-problem that if it is overcome, can rid the designer of its initial design problem. In computer field, such resolution method is called brute force. No algorithm or heuristic help to search in the solution space and the designer, wait at the end of process to see the result of his creation : the design tool. The design tool, perform the design remotely and the designer, at the

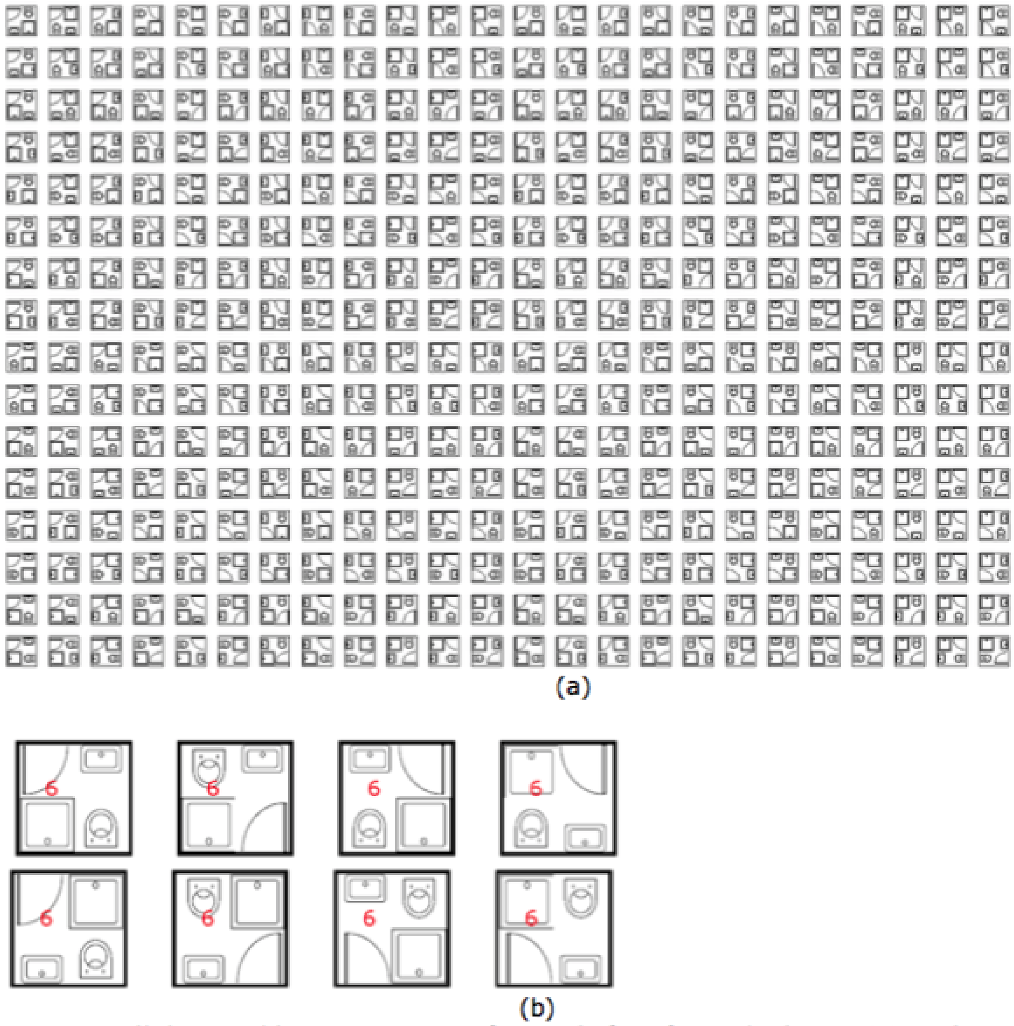


Figure 2: All the possible arrangements of a simple four-fixture bathroom (a). These non-repetitive, rotationally-specific arrangements are 384. However, after eliminating all arrangements that have a toilet seat facing a door and eliminating any arrangement that uses more than 6m of pipelines (i.e. choosing the least expensive ones) the number of successful bathrooms is only 8 (b).

FIGURE 1.7: Variation of all the 4 unit possible bathroom. *source : Kostas Terzidis in permutation architecture presented at ALGODE 2011Symposium, Tokyo, Japan*

end of the chain choose between only for 4 bathrooms[3]. The brute force is not the only computer thinking feature of this process, indeed, the all method of setting the problem with a design space that can be encoded with 4 bits is itself inherited from computational thinking approach. As the sociologist Lucius Burckhardt[40] pointed out, when designer add such sub design requirement to their task it can be reproached that they intuitively, and so arbitrarily, solve by reducing to "*so-called essential* the problem. This is done at the cost of enormous sacrifices on the problem definition for a little value added. Indeed,

[3]We can note that according to Gänshirt categories, choice, even between 4 bathrooms, is made with a verbal tool of criticism.

the by product of this added design requirement through a specific methodology is the insurance of having tested all possibilities of possible minimal bathroom of 4 units. It can be seen as a very small counter part regarding to the restriction of the brief. Since it is an experiment meant to be a proof of concept, this critic doesn't suit. Nonetheless, on a more general aspect, the computational thinking is itself embedding this flaw in its essential prejudice. Indeed, the computational thinking only is maybe not enough to cope problems that are out of computer environment[41]. Some advocate for a *system thinking* instead of merely relying on the faith that CT will be a panacea to all design problems. The author advocates that designers are smart enough to know where and how use this way of thinking and that making the economy of using it lead to a loss of computer as design tool.

The book used to present this brute force based tool provide the code for pedagogical purpose. By this open source attitude, the author provide a derivative by product that may have more value than the initial one. More than the sole tool's code, the methodology can be copied, improved and used in other cases. Situation where it may adequately fit to a design problem without the need of too much reduction of the problem. Unlike personal "methodology" based on analogical tools, or analogical use of digital tools, this methodology can be exchanged in a "solid" form. As all intellectual product, its non material form allow the hybridisation with others. This contribution to the design methods research field is probably more valuable than the proof to chose amongst the 4 viable bathroom by testing all permutation possible.

1.7.2 Computational design by breeding

Another ongoing project can illustrate a CT approach of design through the construction of a tool is the *arkhitektome*[42] project. In its initial phase it may look very far from the concern of real design problems but the version presented here is selected for its abstract quality that helps to understand the computational concept rather than focus on the architectural solution. More recent version of this research project show results closer to architecture concern but for the ease of the explanation the reader should keep an open-mind on the definition of architecture problem. The *arkhitektome*(Figure 1.8) project takes its place within a broader context of creating the sum of all possible architecture. To tackle-down this goal Shoei Matsukawa and the members of his lab are conscious that a brute force strategy is inappropriate regarding the size of the solution domain. Testing all possible shapes and eliminates all the non-viable ones is an unrealistic computational challenge. Above that, the binary encoding of a shape grammar that can cover all architectural expression constitutes maybe the biggest architecture design and theory challenge of this enterprise. To cope with the travel in the solution domain, they decided

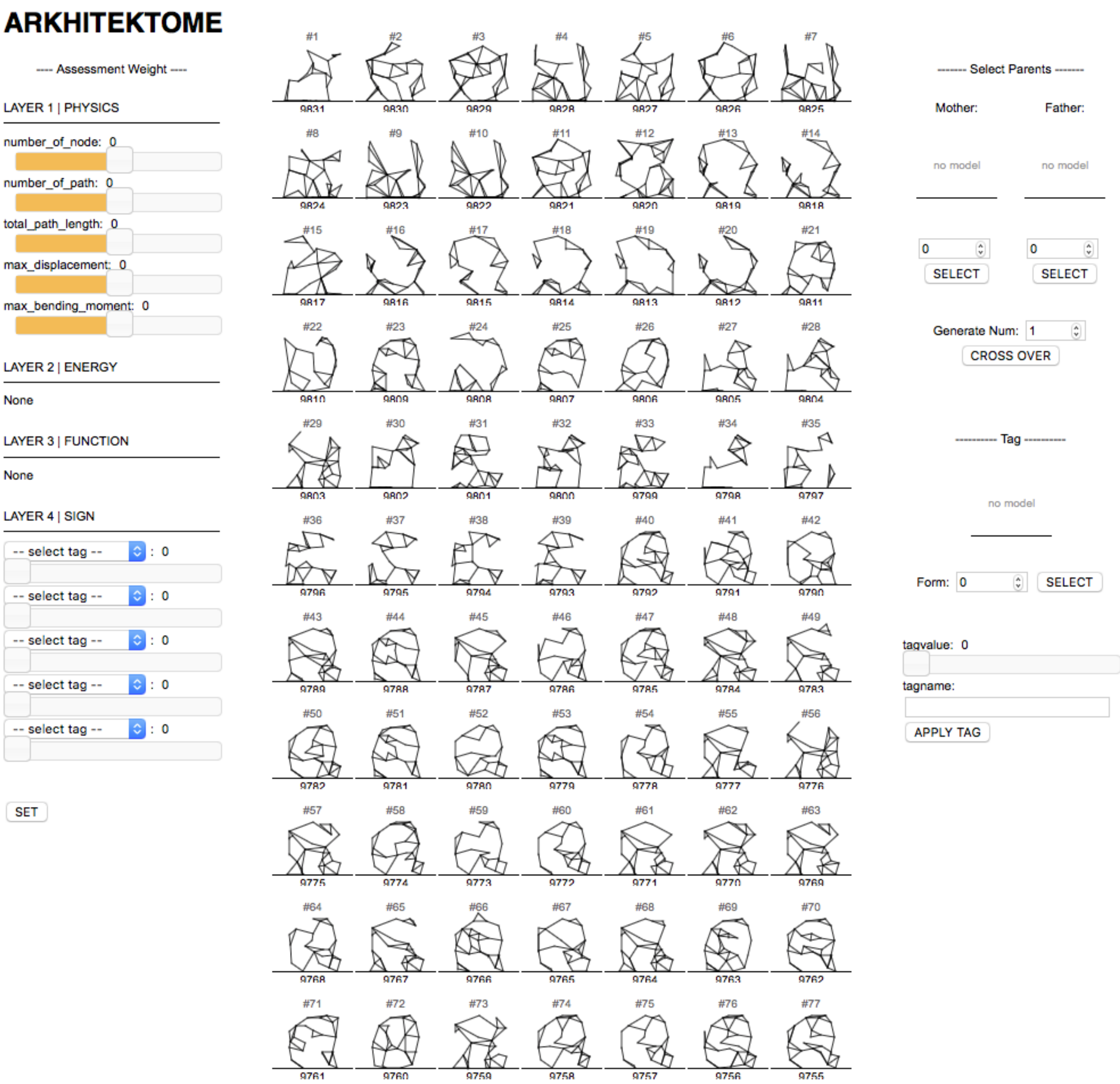


FIGURE 1.8: The home page of the system show the last creation of the system *source : Shoei Matsukawa / 000lab in Arkhitektome©(2013) at* `http://arkhitektome.com`

to rely on computational strategy : Genetic Algorithm. Paradoxically, this strategy is inspired from the explanation of nature by life science and it sounds weird to call it a "computational thinking" approach. Nonetheless, the necessary *decomposition* in sub architectural elements is typical of a "computational thinking" approach. The idea can be resumed as breeding architecture representation on the model of the human coupling. From the coupling of two architectural representations, the system generate children that share and mix features and traits of the two parents. On the genetic code mixing model, the "code" of both representations are mixed with random mutation in order to create diversity amongst the litter of representation(Figure 1.9).

Since the representation is quite abstract (Figure 1.10), they decide to tag representation

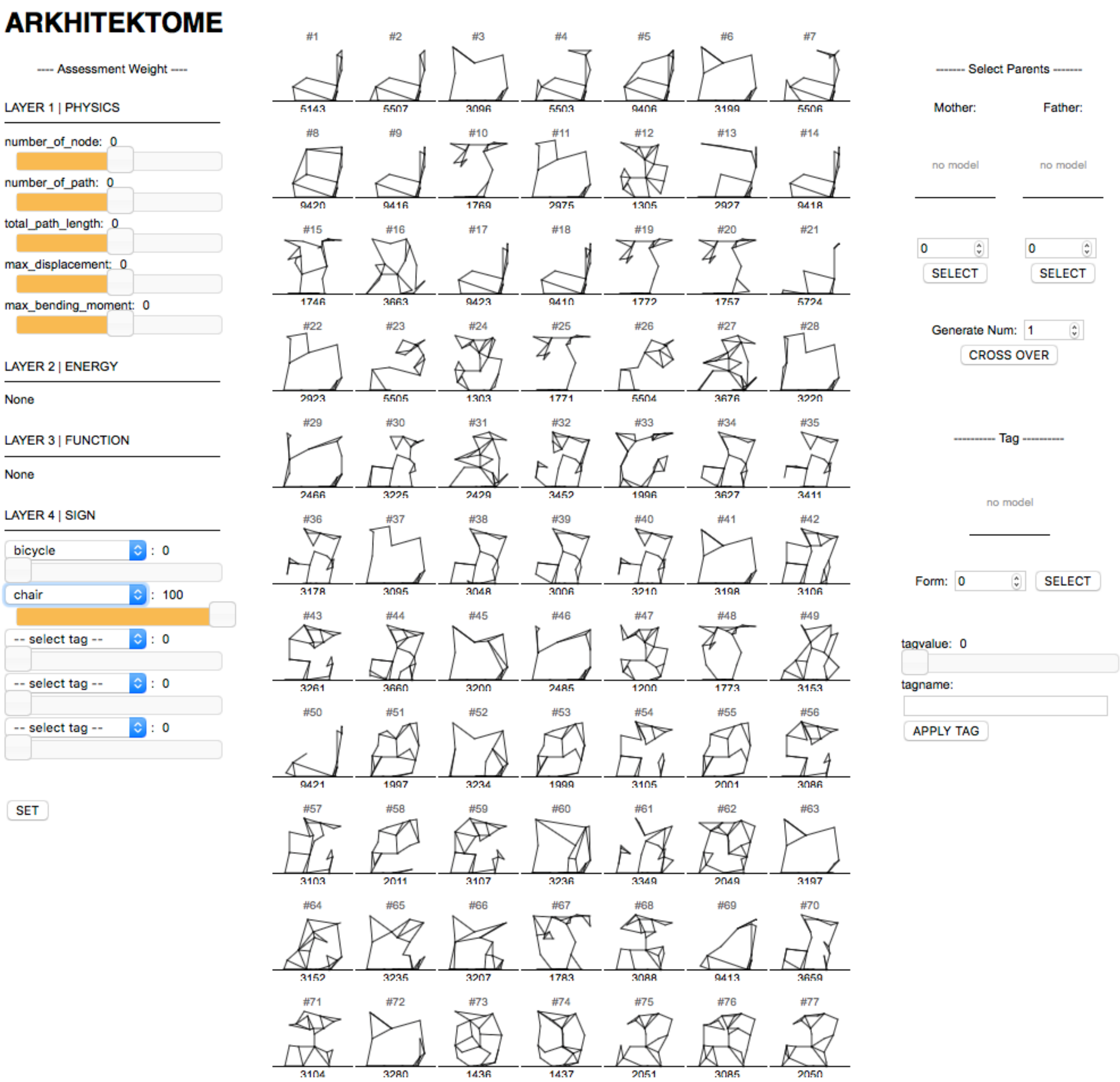


FIGURE 1.9: All shape generated by the system that have been recognized as chair, some get their "chair" tag by inheritance. *source : Shoei Matsukawa / 000lab in Arkhitektome©(2013) at* [*http://arkhitektome.com*](http://arkhitektome.com)

([Figure 1.11](#)) according to the meaning they project on it. The system autorize multiple tag for a same representations and the tag system is "just" a way to navigate in the sea of possibilities with a semantic entry point. As we will see latter, this idea is opening deeper layer in a computational thinking. Through the interaction with the user, the system create a *volatile* value that remain between the system and users. This attempt of control open the last topic that intersected with all the previous topics creates the studied phenomenon.

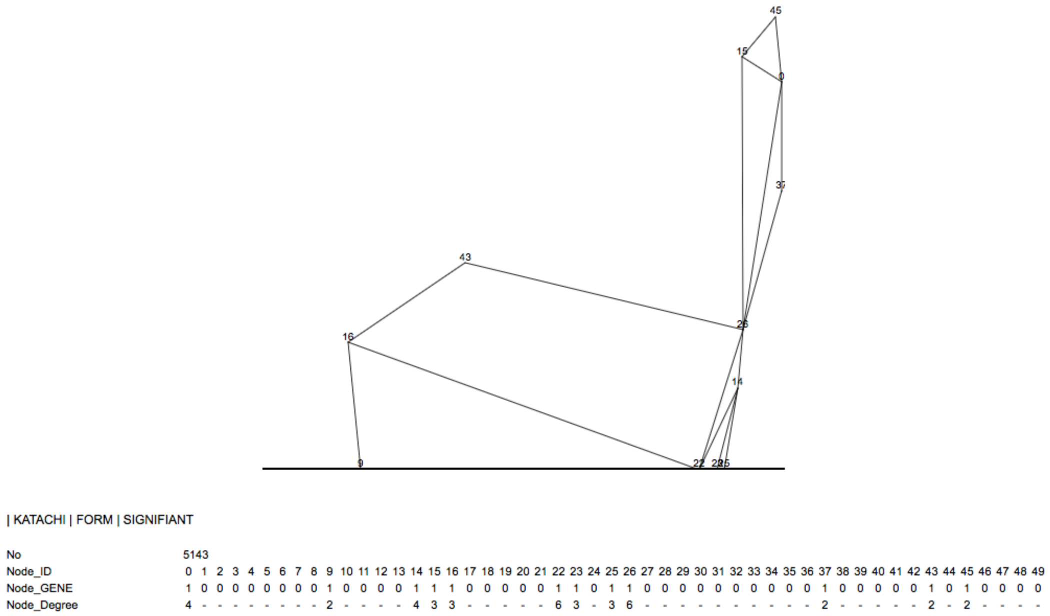


FIGURE 1.10: The "best" chair of the database of Arkhitektome, or at least the one who receive the most often the tag "chair". *source : Shoei Matsukawa / 000lab in Arkhitektome©(2013) at* http://arkhitektome.com

1.7.3 Cybernetics, human in the loop and intervention point

Indeed, this system could work without human intervention for selecting the representation to cross-breed and tagging the output according to their constructive perception. The system can randomly select pairs of representation and tag them according their ancestors lineage. Each shape generated will the grand-grand-grand son of some primitive initial shapes generated. As a "selection criteria" the system integrate a primitive bi-dimensional "moment force" calculation that can be used to decimate non viable shapes. Based on this criteria the GA can even be set to boost a path of evolution decide by the

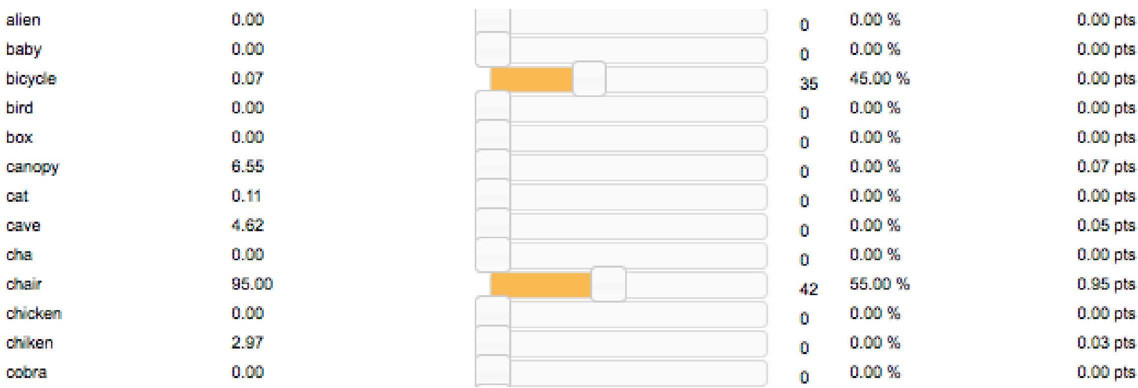


FIGURE 1.11: Different tag that have been attributed by user. In some way this data base of tag is a form of knowledge of the perception ability. *source : Shoei Matsukawa / 000lab in Arkhitektome©(2013) at* http://arkhitektome.com

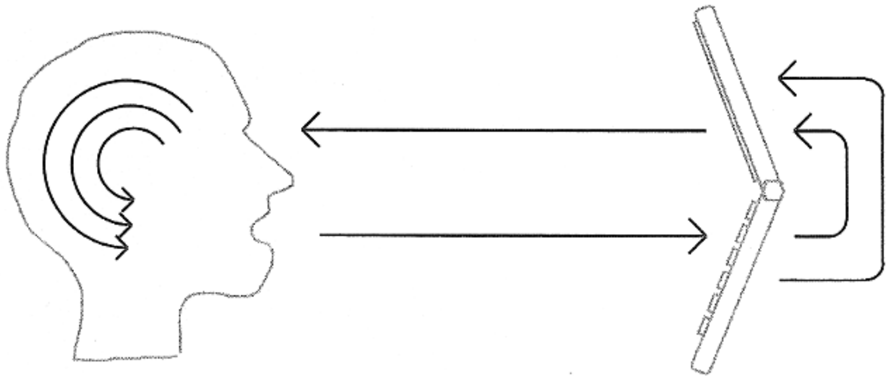


FIGURE 1.12: For Kostas the algorithmic design is about discovering what we had before.*source : Kostas Terzidis in Algorithmic Architecture book[16]*

user at the initialisation of the system. Nonetheless, the sea of possibilities generated by this way have a low value since the access to the suitable shapes after several breeds generation becomes itself a problem.

Putting a designer in the control loop allow the indirect encoding of their feedback within generation made by the system (Figure 1.12). Indeed, something surprising in the system is the navigation in the sea of possibility through the semantic label system. When cross breeding a shape that look like a dog with a shape that look like an egg the breeding produced look like a mix between the both. Despite nothing about the shape of dog or the shape of egg have been explicitly coded in the algorithm, the feature of the representation that trigger the naming by user of the shape remain in the mixed code. Initially, the cybernetic, this science of the control, is not specific to the computation. Control loop feed-back exist in natural phenomenon and they were widely studied in question of missile ballistic that while using computation for calculation the trajectory of the missile, it was not intended to be a design method. It is more a by product of an imperfect control than a creative device. The design process, with its cycle of representation and conceptualisation can be seen as a control loop feedback too (Figure 1.13). But in this case, their no really any antagonist force to regulate trough incremental modification of the settings.

1.7.4 The Turing test, an interaction test

The A.I. present a strong hypothesis that computer with the right program and architecture may think and attack general problems. The first expression of this idea was proposed by Alan Turing in his paper : imitation game [43]. He proposes a "strong" intelligence test for "thinking machine". This test was basically a computer that deludes an "average interrogator" during a "chat session".

Lots of critics have been made to demonstrate that test doesn't prove intelligence but ability to imitate non intelligent behaviour of human. The test itself is not very efficient to determine intelligence but it was, in 1950, introducing a bold idea of machine thinking. Originally, Turing was trying to express what could be the meaning of "thinking" for a machine. He proposes an indirect measurement that was even not aimed to be a measure of intelligence or ability to chat like humans.

That's why it will be unfair to critic Turing proposition 70 years later after for the flaws of his intelligence test for machine since it was aimed to promote the range of machine capacity more than propose an effective test. At this time, nobody had such vision and most ancient programmer, Ada Lovelace, had a very pessimistic opinion on that.

> *Computers can't create anything. For creation requires, minimally, originating something. But computers originate nothing; they merely do that which we order them, via programs, to do.[21].*

Indeed, the seminal paper of Turing point out the need of a model of intelligence in order to determine if machine can think. The interrogator know he is maybe chatting with a computer but there is chance that a human is on the other side of the computer. Nonetheless, the conversation program in the computer should be sufficiently refined to convince the interrogator by various strategies. Those strategies were imagined as similar

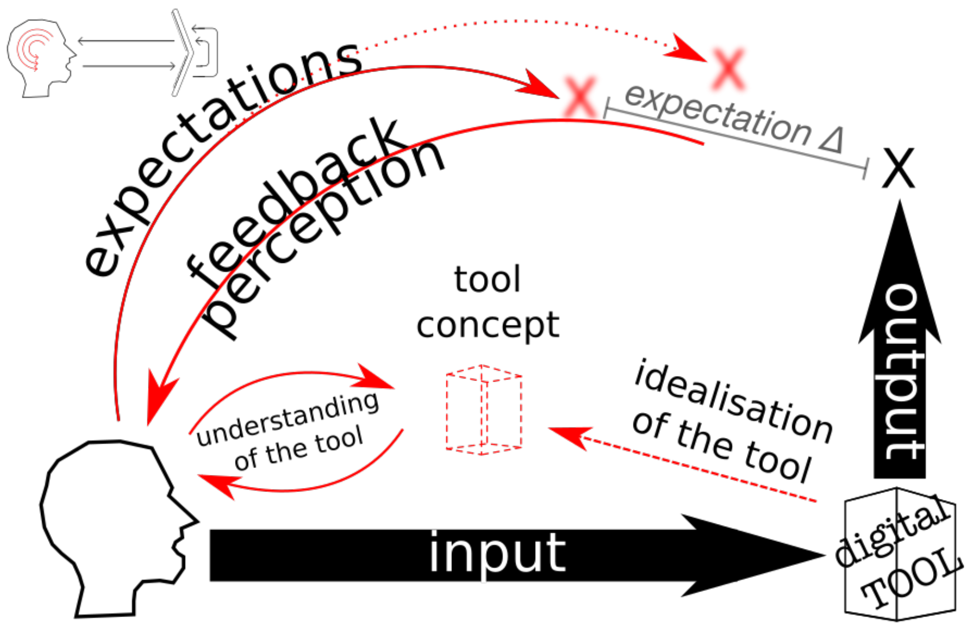


Figure 1.13: The analysis of the control loop feedback proposed by Kostas shows a complexity of interactions

to the one developed by man, ruse, imitation and argumentation, pretending being woman in the imitation game, the social game that inspire the Turing test. These imitations and ruse triggers in the observer an obvious feeling of recognition of his own abilities in the conversation produced by such A.I. Despite it is just a feeling, this indicator of intelligence shouldn't be despise since it is our sole criteria for notions imprecisely defined such as creativity or intelligence. It stated a powerful idea that is in the background of this research: Intelligence is meant to be recognised by a third party.

The Turing test shows the intuitive idea of searching feature to recognise intelligence. In the infancy of computer science, the excitement triggered by such new fields may have occulted a reflection on what are the best feature to use in order to recognise intelligence. This thesis aims to suggest ways to go forward on the question of artificial intelligence and creativity in tool for design. That's why, this idea of recognition by not an "average interrogator" but a "specific designer collaborator" is a central idea for understanding CR.

The question of testing intelligence in machine is difficult since the intelligence in human itself is hardly testable. Indeed, IQ tests reveal only a type of intelligence [44] and this is not an absolute and timeless reference. The IQ test average and standard need to be re-adjusted every year [45] in order to keep some relevance. The idea of testing computer's intelligence has been initiated by Alan Turing [43] with its imitation game for computers. From that, various tests have been proposed [21], [46],[47] to enlighten what could be intelligence for a computer and in the meanwhile redefine what we consider as intelligence. This paper proposes to consider the intelligence in the machine under its weak hypothesis. Above that, it supposes intelligence as a contextual quality and the measure of it passes through the indirect measure of the proportion of the relation's context shared by both entities. We propose the concept of "Computational Resonance" (CR) to relate this limited aspect of intelligence. This new concept escape from the common issues of AI [48] and Computational Creativity (CC) that search those qualities as substantial to the artefact. Indeed, it escapes to those kinds of critics because the "resonance" doesn't aim to state about truly being intelligent but rather about a feeling occurring in a transparent Human-Computer Interaction (HCI) situation. We limit the study to the observation of interaction in order to underline the common context and measure it.

This notion of CR is related to AI in his weak hypothesis. Indeed, AI, in his weak hypothesis present a relative utility in design if the scope of the task is too specialised and rare. But for general tasks requiring autonomy, constructing intelligent machines between the superhuman level and optimal level [49] present a clear interest for design practice.

At the superhuman level, it can reduce the workload of the designer by conducting autonomously sub-task of representation and at optimal level it could even solve design sub problems with certainty. Nonetheless, what Simondon [50] calls the "indetermination margin" of the machine still needs to be managed by the designer. "Indetermination margins" in digital design practice is the linking of digital tools, data, AI, algorithms... For designers involved in Computational Thinking (CT) approach[9] of the design practice, "managing those margins" can be an expression to summarise their design process. Their practice is usually called "Algorithmic design" or "Algorithmic architecture"[16] and aim to computationally "generate" the design proposition. Such CR measurement takes its relevance for those practitioners and theorists.
CR in the strong hypothesis of AI, could aim to more demanding test[51]. Indeed, a super human level of AI suppose that even those margins don't need to be managed by human any more, at optimal level it would be the end of the design as we know it and the beginning of apocalyptic sci-fi movies. But in this thesis, the CR is at a very elementary level of intelligence. Above that the scope of this intelligence is limited since it is restrain to the relational context of only one users involved a interaction with one digital tool to conduct a design task in collaboration.

1.8 Computational resonance

The name of this concept have evolved from "Artificial Creativity Couple" to "Computational Resonance" in order to convey more clearly the intuitions spoted during the later research background. Indeed, theoretical thought and experimental project presented here seems to converge toward that the tool, and more specifically the computational tool can play a new role in the design struggle. This role is a role of companion for design. This "xeno-Intelligence", an intelligence of another type, came to complete designer's natural intelligence. As "intelligence" in computer is the translation in algorithm of scientific knowledge, various type of relation can emerge depending which "computerised" scientific knowledge, which intelligence, is connected to the designer one. The idea to connect the science of design to the designer through the mean of automation of computer leads to create a specific "couple" that may enter in synergy for creative or intelligent issue.

Connecting a science of decision making to computer science through the field of artificial intelligence lead to attempts to provide digital tool for design. Simon was one of the pioneer in 1956 to create this link and meets the ideas of Turing but with another concept of Artificial Intelligence. An intelligence to support human mind and solve bigger problems instead of deluding it through imitation. This approach to design an

A.I. play a strong influence on the development of the field toward the weak hypothesis. Indeed, two program have been created : the logic theorist[52] and the General Problem Solver[53]. Other attempt have been made to link computational tools and design in order to tame the complexity of wicked problem. Rittle developed IBIS (Issue-Based Information System)[54], such a tool to help taming wicked and it founds its full potential when it becomes a computer software such as Compendium.

But the idea related in mind theories and the strong point of the project presented in this research background can't be summarised as just a question of connection with the computer. Indeed, worldwide lots of designer are in a specific relation with digital tools but they are not of the same nature that the one described through the CR term.

The keyword of "resonance" have been chosen to refers to resonant frequencies which provide a response amplitude at a relative maximum. This metaphor illustrate the "maximum amplitude" that can be reach if the correct "frequency" is set. In the case of the system: digital design tool - designer; the conducted research aims to define the parameters of this frequency. At some point the metaphor should be left and the frequency of the system is not summarised to a quantity of energy per second like real electromagnetic waves for instance. Nonetheless, the idea of oscillation present in the concept of resonance have motivated the choice of resonance instead of "couple" that may only convey a static idea of this core concept. Above that, the term of Artificial Creativity has been dropped since this "optimal" collaboration can be related to an improvement in Intelligence or Creativity or another quality that the designer may "recognize" during the interaction with a computational tool. This resonance happens has much in the designer than in the computer, in fact it happens in between even if only the human side can recognise. It happens as much as the human conceptually travel to the computer's logic than the computer is bring to the human logic. One way is not better than the other and in fact its often half of the way that is travelled by both side. Computer Resonance happens when the both meet on the path. It is not limited to design situation. This case can be illustrated by this little anecdote about the logic theorist program that proved 38 of the 52 theorems of Principia Mathematica [55] written by Bertrand Russell and Alfred N. Whitehead. It happens that one of the demonstration was more "elegant" than the one produced by the two original authors. When later, Russell, who has no specific background in computer science, saw the alternative proof, it was probably a case of CR. Logic and advanced mathematics is the natural "context" of Russell mind, the logic theorist program produced something "intelligent and/or creative" in the context of the observer : Russell. The legend says that he "responded with delight" to this surprising creation.

This CR phenomenon is not limited only to ConvNet based tools. We have the assumption that during interaction with any digital tools that contains AI components a situation of CR can happen. What we call "AI component" is defined by a part of the program that provides a form of autonomy in adaptation to complete a task that is related to the weak hypothesis of AI. Unlike the strong hypothesis of AI that proposes that machine may be able to actually think, the weak AI hypothesis limit the scope of AI to limited autonomy and average human level performance to achieve specific tasks recognised as requiring intelligence. Such task can be, making the conversation, logic demonstration [52] , playing chess [56], go [57] or jeopardy [58]... To summarise, task with a restrained and clearly defined context. The weak AI will simulate intelligence in situations where intelligent behaviour is commonly well grounded and its expression anticipated. Indeed, an AI which is made to win an asymmetric game will not be intelligent enough to detect flaws in the game rules that impeach its victory. This example shows that weak AI is defined mainly by bounded context. Usually, such AI are tested on their performance. Thus, game situations propose ideal testing condition : winning the game is a proof of intelligence. But little have been done concerning the interaction with AI. For instance, Centaur chess is a case of interaction with AI focused on the collaboration : "An interesting possibility which arises from the 'brute force' capabilities of contemporary chess programs is the introduction of a new brand of 'consultation chess' where the partnership is between man and machine."[59]. Thus, the pure evaluation of computer chess intelligence becomes obsolete and the collaboration as interaction form becomes prominent.
Base on a similar approach, this thesis proposes to evaluate AI in "Centaur design" condition. Centaur relation in the game and design situation can't be evaluated in the same way. Games embed an evaluation through the winning condition and the competition between opponent. This kind of evaluation is in fact looking on the overall performance of the couple man-machine. Unlike this evaluation paradigm, the CR evaluate the relation itself. The paper proposes to observe the machine and human overlapping context part in order to get a clue of the CR condition of appearance.

CR is the first sign of shared mental models between the user and the tool. For instance, in the use of Neural Style algorithm, the CR is the pareidolia activated by the computational generated stimulus. From this specific phenomenon rooted in the perception process a coupling in the sense making is nascent. The resonance is not the coupling itself, because such coupling can be build upon various process. The resonance is the sign of a nascent coupling on the basis of an unexpected artificial experience of the world from the tool. The resonance term is chosen for to express how the encoding of this experience of the world emerge during the perception. In the ECM, this encoding is matching the perceptual logic without any effort of re-actualising the subliminal mental model. The digital environment match the subliminal mental model in through a shared reference to the world.

Chapter 2

Design process involving digital tools with potential for Computational resonance

2.1 A role for digital tools in the design process

Has seen in the previous chapter, in a context of architectural design with digital tools based on computational thinking approach that place the user in the loop of control feed back may be understood as specific design practice category that has not been studied yet. Indeed, study on digital tools focus on teaching methodologies[28], tool practice methodologies in between tools utilisation teaching and digital design organisation methodology[37]. Or they may be design methodologies, or style, that promotes computational approach[16][60]. The analysis of the singularity in design practice that merges digital tools and mind conceptions in order to observe the effect "on the user" of the encounter of Meta design through process design, cybernetic, mind and design theory have not been conducted or have not been found by the user during its extensive research. It may be questionable that so rare and specific case may present any value for designers in their real practice since the two cases presented before were not in a real design context and seems to require skills that are not usually taught extensively to designers.

2.2 Overview of the digital design toolbox

Design was the pre-scientific way to solve problems and to take actions on the world. Designers didn't need a science of design neither computer to realise masterpiece and the teaching by references and example were sufficient for perpetrate the ancient art of building, of solving temporarily the tension between nature and humans that despite being part of it creates through the artifice a mediated way to disconnect while reconnecting through other means. The artificial accumulated by human civilisation constitutes a culture that is protected by humans from the natural and irremediable corruption. The human being has now a new challenge of being the link between the natural *already there* and the artificial *created*. One is eternal and the second temporal, the game is to keep "alive" as long as possible, a culture that does not have the means yet of its self-sustainment. This challenge is called now sustainability but it can be reframed of a more general context of taming wicked problems. Indeed, while the sum of knowledge is increasing, the sum of unknown increase exponentially[1], more and more informed designer get conscious of losing the control of the consequences of their choice. To at least, keep operative the remaining tangible knowledge hardly produced by scientists, the designer needs to automatise proven part of his design in order to focus on the uncertain part that will be treated through the design iteration. Since science modelled mathematically can be translated in algorithms, through simulation and automation of computation he can get relieved of a part of the task due to design iterations. Then, the next level is to automatise design choices that are depending of parameters that can be produced with scientifically based algorithms. Compared to the simulation that aims to be a solidification of proven knowledge about *nature*, automation of noncritical choice is what can be so called A.I. in it's weak hypothesis, automation of the knowledge about the *artificial*. G.P.S[53] and the logic theorist[52] are such a form of solidification of the proven part of design. The digital design toolbox lacking of such tools and the means to connect them.

2.3 Ready made tools

In all the range of digital tools that can be created or that are already accessible, can we find some features or trait of CR? To address that question, the author think that could be useful proceed with method and answer to it for three categories of tools. As we will see later, everything is not about the tool but about the relation between the designer and the tool. The presence of CR is depending on the operator and the way he uses the tools. Depending on how things are used, in design practice, everything can be a tool. Such assumption made the answer even more difficult to grasp if categories are

[1] `http://kk.org/thetechnium/the-expansion-o/`

not created. Three categories aim to cover all possible tools and their various uses. The ready-made tools are the tools that are advertised as design tools. They are sometimes literal metaphor of pre-computer design tools that have been adapted to a computer interface context. Then, more focus on the tools and all their use, the hack of tools as a source of CR will be questioned. Finally, the case of building his own, an already proven source of CR signs through the example of *arkhitecktome* and *permutation design* will be studied in depth in order to present how digital tools can be created. This part aims to provide a computer culture to the reader in case this one doesn't match with the author's.

Despite tools such as the *the logician theorist* or the code of permutation design are "already made" they are not considered as ready-made since they are hard to access for novices and they are in somewhat too specialised to their creator. We will see in the hack part that this over-specialisation can be overcome but for this part, digital tools concerned belongs to the tools that share family resemblance with GIMP, AutoCAD, 3Ds Max and other similar tools.

How ready made tools are made

Computer software are interface that control computation in a predetermined way. Professional programmers belonging to a company encode a computational process that may fulfil the need of users. The determination of those needs is decided by marketers in software companies, but by contrast in open source projects, they can be defined by any programmers that possess the skills to implement them. If those needs definition engage the software development in a path incompatible with other software development plan, the development is forked in a different branch that will be maintained by programmers that recognise value in it. The case of open source development shows the originally free and malleable aspect of digital tools. This freedom has a cost in skills that need to be developed by the designer. That's why by convenience designers look toward the software market to select tools for their design process. Those software are marketed as tools for design and bring forward the productivity as the surplus value. Implicitly sating that the creation is the designer's business and that occurs in his mind. This vision of the tool differs slightly from Flusser we've seen in the chapter 1. Indeed, they are production tools and they can be used without any interaction that may make evolve the initial intention but by their sole knowledge as possibilities of production transform them in class of operation in designers' mind. This class of operation, that we call digital operations, reach a similar status as operation of conception in the conception space. This case is the typically a case of digital tools "corrupting ideas" of the designer as

Gänshirt[25] mention it. In this case, it is even a corruption of the thought process itself, the designer that makes this assimilation may have forgotten the context of his problem and tries to solve them with unrelated tools. If digital designers are unaware of it, the software offer of tools becomes his sole and unique tool kit for digital operations. Despite some conversion format or export and import missing features restrict the possible sequence of digital operations. Usually, within one software, the designers have at least the guarantee to be able to freely move from one operation to another. That's why software are a reliable source of tools and the interoperability of these tools is a weighty argument to choose a software amongst others. The ease of transition between operations incline users to stay within the proposed boundaries.

In the end, designer can through internet access to a large variety of tools. Within this variety of tools, we can delineate tools that are advertised for production and tools for design. Tools such as GIMP (Figure A.2) are presented as edition system while PhotoScan (Figure A.8) is a production system. But if we look production tools, their outputs are always up to the validation of designers. There is no production for the sake of production in design. During the design process, everything produced is *making for* [36]. Thus, it means that a creative and a productive quality can't be delineated in tools. In the other hands, there is no tool purely made for creative exploration without the possibility to save some data for later production.

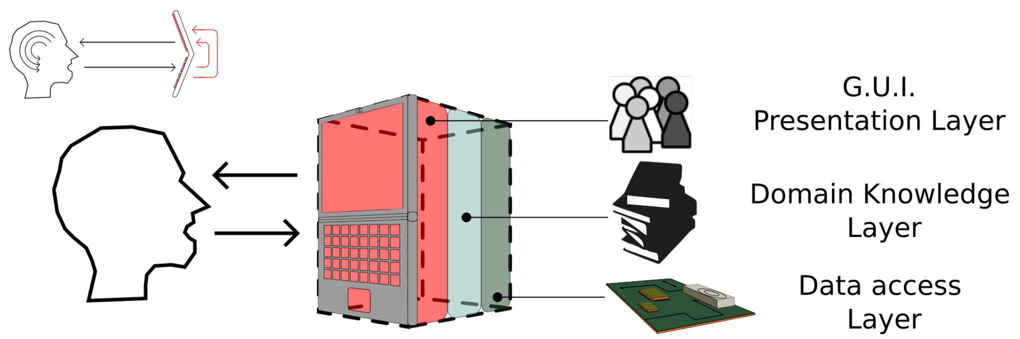


FIGURE 2.1: Software architecture is usually composed of 3 layer, two external layers connect domain knowledge layer encoded by professional to the environment of the software : the User and the Hardware

The case of Al.chemy show a rare case of a software with a strong orientation for the sole creativity.

Standing from a different point of view, all tools in a computer are editing the data in a computer and such distinction between creation and production doesn't make sense for computational designers. They look for any means to edit the data, to transform it from one shape to another.

2.4 Hack of tools

The extensive examination of all 'Creative' use, of tools that are not initial made for design is unrealistic since it means analysing all existing computer tools. Nonetheless, the example of Al.chemy show us that tools that let a large role of interpretation for the user may find some potential for CR. Regarding the extended definition of tools provided before, even if belonging to the category of hardly accessible tools, the algorithm of Deep Dream[61] and Neural Style [62] are "research tools" that have a hack potential for design. Indeed, initially created for studying what is encoded in the ConvNet, they can be used to produce images that maybe use in a design process as inspirational material for instance. By some hack of the type of the expected input, they even can become tools that are located in a digital design process. Compared to Al.chemy[63], we can say that they are of a similar type since they produce output that wait for constructive perception[35] from the designer in order to reveal their value in the design process. Even if it is not in the same way since the interaction process is asynchronous and the input way is indirect, we can notice a similar design attitude that may lead the designer to select such tools. The designer may select this type of tool rather than something like Al.chemy for the input way. Despite the input way of the both tools seems to be visual, the ConvNet based tools are indeed from the verbal category. Al.chemy main feature is its connection to gesture that made this tool belonging to the visual category.

2.5 Building owns tools with other digital tool

For designers, there are different ways to build their own digital design tools. They can in the purest fashion start from scratch and try to use the programming language the closest to the machine. We can see an example of such extreme approach in the development of MINUIT[2] but more mainstream approach rely on libraries, high-level programming language and interface with other design tools. At least, even the most extreme approach doesn't go to the point of creating their own language and thus their own compiler. The compiler is like the Swiss army knife of the developer in the sense that lost in desert island – an empty computer, it is at least the most fundamental tool that he needs to take with him in order to be able to design build up other necessary tools.

But it is possible to create tools more easily if we admit a more flexible definition of digital tools. Indeed, if a set up of a chain of tools allows to produce, and so design

[2] http://www.rvba.fr/web/index.php/projet/minuit

through the adjustment of the production's parameters, it can be considered as a self-made digital tool. The example of the Belgrano[3] workshop illustrate such type of loosely home made tools. In this case, design this tool result in the design of a re-enacted design process. We could name this tool a "function based shape design scheme for laser cutting production". A very specific tools that can allow users to design especially those kind of shape for this sole mode of production. But as it was designed, there are lots of entry points where this tool chain can be shifted to other specific design goals.

2.5.1 Libraries, API, Data structure, the component of home-made digital tool.

The process presented in the Belgrano Workshop rely on file interoperability. Thanks to the "*.obj*" of Wavefront different tools in different software can be "connected" in such way that it forms a chain of tools. The "*.obj*" is one of the first 3D file formats and it is widely accepted by various software but this transfer of files through import and export of file is cumbersome. Indeed, lots of "non-design operations" are disrupting the flow of measure attribution required by the created tool.

To create proper digital tools, programming language are used to *glue* various existing components that are not made for interaction through GUI. Those systems are API, libraries, file formats, XML configuration files and other. They allow an access through the code to tools, and through the code, the binding of those tools in order to create new ones.

Understanding clearly the purpose of each component and knowledge in programming are necessary for creating tools in this way. If the design of such tools is reusing component that has CR features, the user may hope to be able to create a new tool with CR. But the case of *arkhitecktome* shows that addition of very functional components that don't have CR qualities can result in creating tools with CR features. *Arkhitecktome* website is based on django, in its 3D version, the Rhino 3D script and all those component are not presenting any sign of CR. Nonetheless we can recognise that the main value of project that bind tools by code instead of operators' manipulation is the fluidity of the utilisation. As we saw with Al.chemy, the key of such software is the rapidity of the feedback. If the process proposed through the tool have some delay, all the interest of intuitive constructive reconstruction disappear. The immediacy of feed back related to verbal or gesture input allow a state of flow that can be achieved in any means by the "tool" created for the Belgrano workshop.

[3] http://ikeda-lab.sfc.keio.ac.jp/home/en/project/2013_workshop-in-belgrano-university-in-buenos-aires

2.5.2 Tool chain

The tool created by merging libraries, self-written algorithm and connecting with raw data input format create a chain that is like a suit of digital design operation that may constitute part or all of a conception operation. For instance, if the tool of Belgrano workshop was one stand alone tool, it will perform a conception operation that simultaneously create a measure attribution in technical, and optical scale. The "meaning" of this operation is "architectural" while the meaning of the intermediary transfer from one a BRep surface to a mesh surface realised in the file conversion at the end of the step 1 have no architectural meaning. Nonetheless, the technical decision taken during the conversion may impact the architectural meaning in the end. So the part of the technical decision that relate to architecture should be extracted and offer to control to the user through the parameters of the tools.

The automation of this tool chain is a condensation of a "sub-design process" related to architectural conception operation. Despite the flow provided by such automation, the system does not present CR quality while their do not show sign of autonomy and common sense in the management of errors. Indeed, the tool, in it's manual recipe from do no provide any handling of shapes or slicing methods that can't be realised. Then, when at the initial step a wrong shape is generated with the Processing code, the user may see the problem generated by the initial choice only later. To fix it, he will have to step back to the initial step of the process. This is a clear lack of CR of the relation with the tools since the way to decide the parameters of this conception parameters don't fit to the way of input parameters in the tool.

2.6 Toolbox as environment

In the previous section the explanations focused on the fabrication of tools and shows that finally, tools are component of bigger tools depending on how they are organised into a synthetic digital design process that aim to help in the measure attribution on one or multiple scale. The sum of the tools known by a designer are similar to the references in architecturology. In a similar fashion that they design architecture, while they design a tool made of other tools in a process, designers select tools amongst those reference in conception operation for digital tool design. To avoid the confusion with the Architecturology, the author propose terms that are similar to the description used in architecturology for describing architecture design but in the context of designing a tool to help to perform a conception operation in architect project.

- The operation of conception becomes automatised process step design APSD

- the measure attribution becomes sub-tool definition S-TD
- the reference becomes existing digital tools, algorithm, environment or Existing options EO
- the dimension becomes the I/O data exchange format
- the pertinence is still pertinence but we call it Digital pertinence

So to rephrase the conception of "digital design tool" with our new terminology : in order to help himself to conduct an operation of conception the designer design a digital design tool. The process to design this tool is conduct APSD operation that consist of defining one or multiple sub-tool (S-TD) by selecting a I/O-DEF then by comparing with the EO the sub-tool is defined according to the pertinence regarding EO.

2.6.1 The initial substrate model, a tool for thought

In parallel to the thesis, the author get the chance to participate to the EDGE project. Since it is a method of teaching entrepreneurship by the practice of a project, I started a project closely related to this research: the creation of a digital tool that content CR feature. Regarding the previous analysis and research, it seems clear that the best way to improve the CR in the digital design is to get as soon as possible in the project the richest substrate model possible. Indeed, since computation is based on the processing of data and if the digital design is about creating tool to perform the design through computation, the more designers have access to data, the more they have option for their algorithmic design. Indeed, algorithmic designers create a process of data that will solve a design problem. To start this process they usually need to input data of the site manually. They vectorize the site boundaries, build 3D model of the neighbourhood in order to use the context in their design process. The effect of this digital tool focus on the first step of the digitalization to create design document. It is indeed the first step of the algorithmic process. If better data can be provided to algorithmic designers, it will increase their design possibilities because they can use more various algorithms if they have access to more data format. To get automated and precise input of data the algorithmic designer should rely on sensors. They produce raw data that should be refined for an algorithmic design process use. The purpose of this system is to refine and convert these data.

2.6.1.1 The starting point of digital design

The system employs an analytic methods using algorithms to conceptualize and digitalize architectural spaces in order to highlight parametric shape. From one group of digital photos, epipolar geometry produce 3D model mesh from high density of 3D points. The mesh constitute a whole indistinct part that differs slightly from our intuitive perception of separate objects that have relative independence. The design process is an alternation of conceptualization and visualization phase. Computational thinking approach of the design involves somewhat a ”computational conceptualization” adapted to an intuitive visualisation of architecture as structured data. Since our perception and ability to act on environment are deeply bounded, an algorithmic control of architectural data will trigger CR. Algorithmic design is manipulation of data in order to solve a design problem. Through caption device, a visualisation of architecture in a digital data form is produced. They are process with computer vision algorithms and machine learning system in order to be refined to a semantic level.

It allows to grab the points through semantic group. Actions can be performed on all the points of the group. When it is relevant, group can be transformed in a more symbolic form. By this way, an higher level of abstraction is reached. Here, it is possible to express relations existing between the group. Through these relations, their parametric identity is deduced. Parametric elements are highly conceptualized type of shape since they can be described like object is OOP : as a process and parameters. By keeping the process and changing the parameters, these elements can be easily modified in a range of likeness. Because the modification of the shape follows the logic of its generation process, this type of design is called : parametric design. It is a prerequisite level of abstraction to do algorithmic design.

The abstraction of unitary architectural elements can be found right at the beginning architecture history. For instance, Greek columns, are ideal shape with logic of generation embedded. Their functionality are common for designer who use it as short-cut in the design process. Those proven solution component avoid to re-invent well proven solution. As Parametric design allow to postpone critical design decisions to the moment where there impact will be clearly understandable[64]. Algorithmic design process is adding a new layer of organization in the design process. With tailored algorithm, critical decision can be taken automatically for optimizing the outcome of the decision. The design shifts toward the construction of tool that will produce the appropriate design. As parametric component, well designed Algorithmic tools can be reused to recompose other design process. This system produce an algorithmic tool for architecturally conceptualize data from the context captured with photographic device. This method could be reused in various algorithmic design scenario like house renewal or contextual approach. In the academic field, it can be used for architecture building study and comparison. At research level, it open the door to deeper analysis of architecture grammar and model. Similar research as the one conducted in the Space Syntax laboratory[65] can be based on the refined data produced by such system.

Capture architectural environment and generate parametrically editable model is at hand. To achieve that, it is necessary to create a system that segment and recognise architectural elements in a digital capture. Architectural elements share the propriety of being designed by human for human. Therefore, some inference tie on the human context can help in the relevant segmentation of the capture. Survey of existing capture system(LiDAR or epipolar geometry from picture produced point cloud) show up that they mainly create mundane 3D mesh that are hardly editable for a creative design practice. The main concern of these software is accuracy for archive and observation purpose. Some software helps to make a CAD reconstruction of the scanned object but it is not adapted for architectural design because they don't take the architect perspective to detect relevant feature.

2.6.1.2 3D capture

In order to improve that part of digital, I started to create a system to capture architectural space with specific device and obtain a digital 3D model made of parametric component and 3D mesh when no parametric shape can be used to express the element. With such tool, it will be possible to articulate inside and outside captured data of the building in order to produce the 3D model as it has been made by an human. This system joins the automation of epipolar geometry algorithm and the semantic embedded in architecture 3D models produce by human. Each pictures provided to the photogrammetry software will be segmented. Each segment will be tagged and a probability of correctness will be attributed to this tag attribution. Then, the tag of each segment in each point of view will be summed up in order to give a solid identification of each part of the 3D point cloud. Based on the identification a detection and modelling strategy will be algorithmically chosen. Each architecture elements will be build with a simple parametric shape. With this data transformation from a set of pictures to the procedure to create a 3D model incorporating CAD modelling logic and structured semantic data, open the way to B.I.M. or more open and versatile set of heterogenous model [66] as digital document useful to start a project. The requirements specification and the detailed explanation of the system are explained in the Appendix E for the encoding form the world information into a 3D modelling format. Despite the project is separated from the thesis, the research made for the development and the thesis research mutually fed each other. Indeed, this system is based on the use of ConvNet and improvement of the

ConvNet practice understanding made during the development clarified some doubt of the theoretical part of the thesis. The system's goal are here mentioned here because it present an example of what could be a tool based on CR.

2.7 Evaluation Methodology of Digital tool for design purpose

After this large overview of all the tooling possibilities with computer, there is a need to organise the tool references in such a way that the characteristics of tools that show some sign of CR can be more clearly defined. Indeed, we saw that two tools based on ConvNet show some sign of CR while that tool chain used during the conducted workshop (Belgrano) was lacking of some rapid and suitable feedback on actions. The case of Al.chemy was showing the potential of rapid interaction that create state of flow if the user had the skills and an ability to have constructive perception on the created feedback. Those features qualities and strong point are disseminated amongst various digital tools and none of them show sign of CR like the example of *permutation design* and *Arkhitektome*. Both were enlightening something unusual in the relation that can be created with their user. Indeed, a key elements of the comparison of such tool is to take

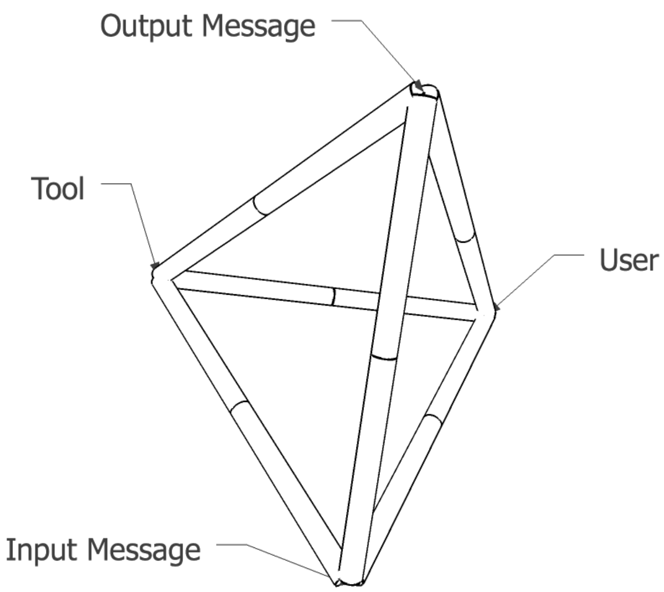


FIGURE 2.2: This system present a 3 Dimensional abstract representation of the relations involved during the use of digital tool

in consideration the relation with the user. To propose a analytic method that can take in consideration all the configuration of the relation in the use or design of digital tool, an evaluation system have been created on the basis of thetraedral relation[67] analysis method by the meta-designer Jhon Wood. Wood's methodology is about managing a limited number of entities when situation is analysed. Indeed the short term memory used for thinking his limited to 4 concepts[68] at time. Thus, it suits to chunk problems with multiple parameters and concept within sub-group of 4 problems hierarchically organized. Luckily the problem of interaction with computer can be formalised with four entities. The four entities of the digital design are :

- User
- Tool
- Input message
- Output message

. From those 4 entities 12 relations can be developed. This method of analysis take all the relations between the entities and analyse it one by one. If we place the four entities on a tetrahedral like in Figure 2.2 we can see that 6 edges exist. Each edge are a relation and can be taken in one direction or the other. Thus we have 12 relation to analyses. To avoid overcoming short term memory limitation, The 12 relation are gathered in 4 group of 3 relations. By this way each group called Orthogonal View (OV) can be studied as a separate concept and in the end the 4 (OV) concept are used to model the HCI in a short term memory friendly formate.
We propose 4 Orthogonal View (OV) term as :

- User-Input-Tool as Production (POV)
- User-Tool-Output as Feedback (FOV)
- User-Input-Output as User Process integration(UPiOV)
- Tool-Input-Output as Yield (YOV)

Present evaluation system based on the analysis of relation involved in design process using tools. The evaluation system contains 9 relationships (Figure 2.3) that suffice to rate the relation between tool and user in all their aspects. In fact, the 3 missing relations in the unused OV can be used for benchmarking between tools that look like each other but they are not useful for the present demonstration.
We propose name for each relations in their respective OV context.

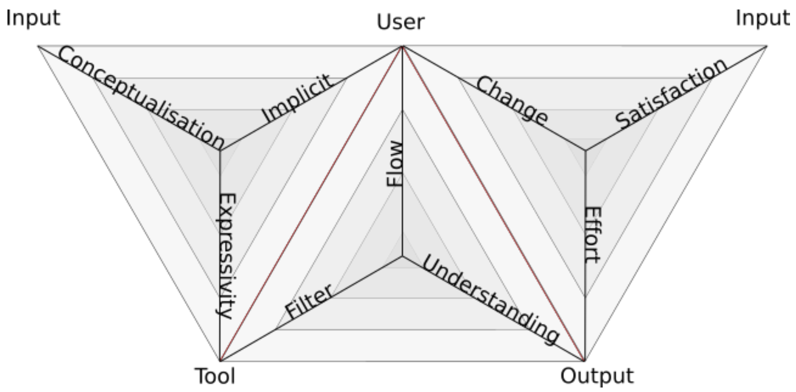


FIGURE 2.3: Each triangle contain three criteria that evaluate one aspect of the tool use

- in the Production Orthogonal View (POV)
  - Expressivity
  - Implicitness
  - Conceptualisation
- in the Feedback Orthogonal View (FOV)
  - Understanding
  - Flow
  - Filter
- in the User Process integration Orthogonal View (UPiOV)
  - Satisfaction
  - Effort
  - Scale

Those nine relations are located on each bisector of the 3 triangle of the Figure 2.3. This Figure represent unfolding of the Figure 2.2 tetrahedral. The axis of the bisector is the scale graduation of the orthogonal relation on the triangle side. For instance, the Tool-Output relation in the "User-Tool-Output as Feedback OV (FOV)" get is measure on the bisector that pass through "User" vertices. According to the Figure 2.3 the concept that explain this relation is the "flow".

Relative comparison system.

The evaluation procedure follow the same intention to limit the short term memory workload by proceeding evaluation by group of four. Indeed, if we ask to testers to

Tools/ relationship	symmetric pen	DeepDream	Neural Style	Fluxus Cube
Expressivity	1	2	3	4
Implicit	1	3	4	2
Conceptualisation	4	1	2	3
Understanding	4	2	3	1
Flow	4	1	2	3
Filter	4	1	2	3
Satisfaction	4	2	3	1
Effort	2	3	4	1
Scale	1	3	2	4

FIGURE 2.4: Ranking of a set of 4 tools by a tester

compare digital tools they use in design without explaining a criteria of comparison we may face comparison made two by two with verbal explanation of the comparison criteria. It is easier for comparison of data to get a numerical form of data. Thus, a ranking of digital tools that are used by Designer can be used as evaluation task but without a clear criteria of ranking they may alternatively rank by time of utilisation, preferences or other arbitrary criteria. For this reason we provide the relations concept as criteria for their evaluation and they have to rank them four by four in each criteria. It segment the task in simple evaluation steps that can be easily conducted by each testers. Indeed, giving a grade on 10 is easier at the beginning but after few grades in the same category a need of coherence becomes apparent. The first grade are given without any references and the difference of number of points between each digital tool evaluated may not reflect the difference of quality proportionally. After ranking 4 digital tools like : Symmetric pen (in Al.Chemy), Deep Dream, Neural Style and the cube function in Fluxus we get a table like Figure 2.4 for each Designer that evaluate the tool.

Then those "scores" are put on the unfolded tetrahedral chart like in Figure 2.5. Each group of 3 triangles is represent a tool and each bisector is used as a 4 levels scale of each relation. Colours are chosen according to a visual code that help to categorize each tool roughly. It helps to visualise more quickly if there is link between category and evaluation triangle of each OV.

We propose to designer to rank other set of digital tools by group of 4. In order to see how the diagram will look like. Set of tools have been made by associating tools that seems to be similar in some way. Here the list of each four tools group of the evaluation.

1. Symmetric pen, Deep Dream, Neural Style and the cube function in Fluxus
2. Processing cube(), the cube function in Fluxus, Rhino script Cube function and a cube in Blender
3. Symmetric pen, raster pen in GIMP, pen in InkScape and the line in Processing

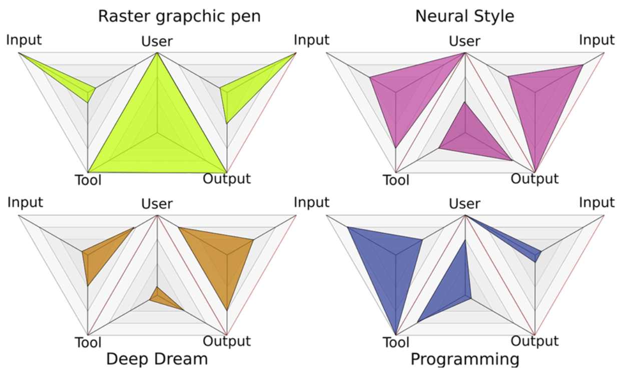


FIGURE 2.5: This diagrammatic form summarize the 3 planes and the various criteria. The center of each triangle represent the lowest value

4. Symmetric pen, raster pen in GIMP, deepDream, Blender Cube
5. 123D make, Blender cube, Neural Style, Processing cube()
6. blender cube, rhino cube, SketchUp cube, InkScape Cube
7. Unfold rhino, Texture mapping Blender, Sketchup line, Fluxus line
8. Deep Dream, Neural Style, Fluxus cube, Processing cube
9. Deep Dream, Neural Style, GIMP Filter, InkScape Pixel Art

If we take the case of the Designer A we can see in Figure 2.6 the synthesis of all the evaluation he made. By looking the same tool evaluated in different group we can establish more complex ranking. Indeed, more tool to rank make the exercise complex. On the other hand, fragmenting those evaluation can create a Condorcet's paradox with inconsistencies in the ranking. Above that, the point is not to merely rank. The ranking is made to create comparison and help to look for patterns. Thus voting paradox are not a relevant critics since this method aim to find singularities of tools and more precisely, found out the position of Deep Dream and Neural Style regarding other type of tool. It is not a uni-linear ranking but to localise two new tools in a network of existing digital tools.

Regarding the result of the experiment shows that too close tool can't be evaluated. Indeed, the comparison of cube drawing function in modeller can't be separate between on most of the criteria. The difference between the tools are mostly due to the relation that are proposed with other tools in the software environment. In contrast, tools that are very different have been easily ordered. From those ordination and the evaluation diagram produced with the result, we can notice that some patterns occur when tools are put together. Indeed, categories of tool present always the same traits when they are

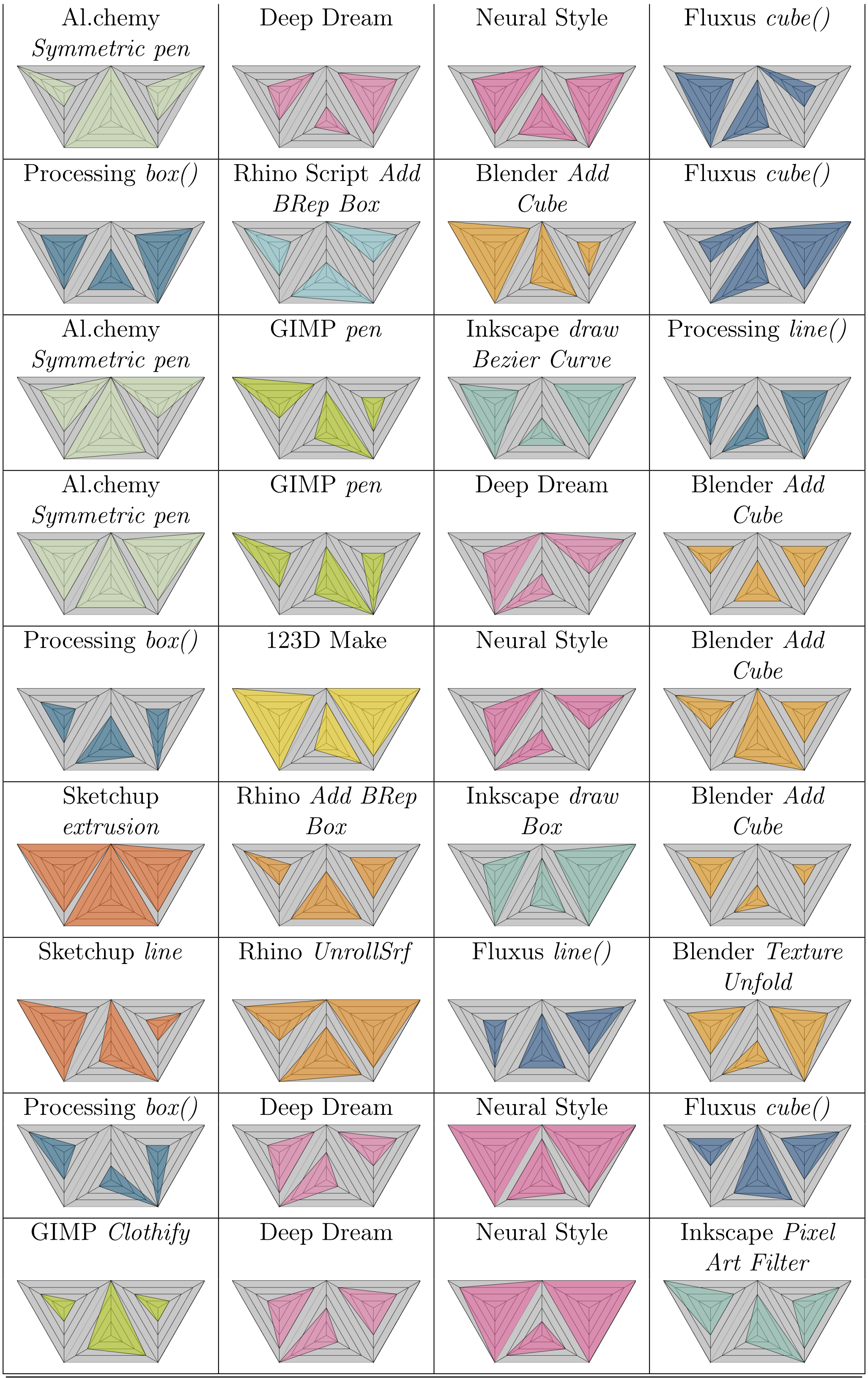


FIGURE 2.6: All group of 4 tools evaluated

put together. When tools of the same category try to be compared together this pattern disappear since it is a relative rating. An exploratory factor analysis of the criteria (EFA)

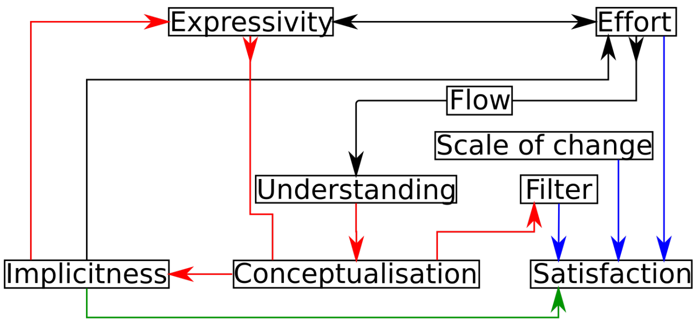


FIGURE 2.7: Criteria have interdependencies some progress in one criteria show improvement in other criteria connected to it

have been done in Figure 2.7 in order to see if covariance of factors can be revealed in order to gather them into testable phenomenon.

2.8 tools typology

When the comparison diagram are looked closely with the reading grid of those relation path we figure that within the typology of ConvNet based tools, users have rarely a clear conceptualisation of the tool at the beginning. It follows that, regarding the communication between relation, we understand that mechanically the user filter criteria is widely open. Above that, the input mode very specific of these tools allow a specific ratio between Expressivity and Implicitness that doesn't occur in other tools in the comparison with ConvNet tools and in other comparison. Another ratio that seems very specific to ConvNet based tools are those that happens in the FOV. Indeed, the ratio Filter/Understanding is similar in the two diagram and if we extend the relation to the Conceptualisation we figure that all those parameters are related to the User-Tool relation in both relations and propose a model of the User that will be tested in the chapter 3.

Chapter 3

Design Attitude

3.1 Design Attitude model

The ranking of tool exercise is done regarding those tool in a design context. Despite it doesn't state a specific situation, the users started to use the tool by setting their own challenge to the tool in order to see what they can do with it. This is comparable to the *problem-finding*[69] of creative experience expect that the context of the problem setting is indeed a creative production or research. In order to *locate* what kind of moment the user is "emulating" with this testing approach, we propose to use the Architecturology vocabulary in order to offer a conceptual framework that is general enough to suit to various design process scenario. Within a design scenario, the full process can be described by following the time line of conception operation conducted to achieve a goal. The digital operation are not the reference chunk of design process. Indeed, digital operation such as install software update, convert a file format into another are not conception operation.

Within a conception operation that may or not use a conjunction of digital tools, the designer will use digital tool to aid is decision process in measure attribution. In case of advanced modelling tools, the measure attribution in architecture is interwoven with a measure attribution in the design of a parametric model[28]. The presented tools for evaluation belongs to two categories. The tools that can only be used as attributing the measure in conception operation of architecture and those can be used for those attribution and for the measure attribution in the creation of a process that will help or perform remotely this measure attribution in architecture. Programming tools such as *Rhino Script* or *Processing* can be used for both. In our case of defining designer's attitude, the meta-design through tool or the *direct* design can be taken in consideration. Nonetheless, we will for the example, suppose a *direct* design situation. To make the measure attribution, the designer may use two different tools for various reason: the first didn't fulfil his expectation so a second tool is used. Or maybe, it is a premeditated strategy because the user know that one tool perform nicely on part of the layout of component and that another one is good to move automatically the component. We can consider that at least, through the choice of the tool there is always a necessary level of premeditation and so, a some expectations. Those expectations are evolving at each feedback from the tool. Those feedback can be the output feedback or intermediate pre-figuration in the scripted tool use process. As we can see in the Figure 3.1, the attitude of the designer is very specific to the timing and we suppose that it may evolve in during time. Especially during the utilisation, the feedbacks have probably critical influence on user state of mind and thus attitude.

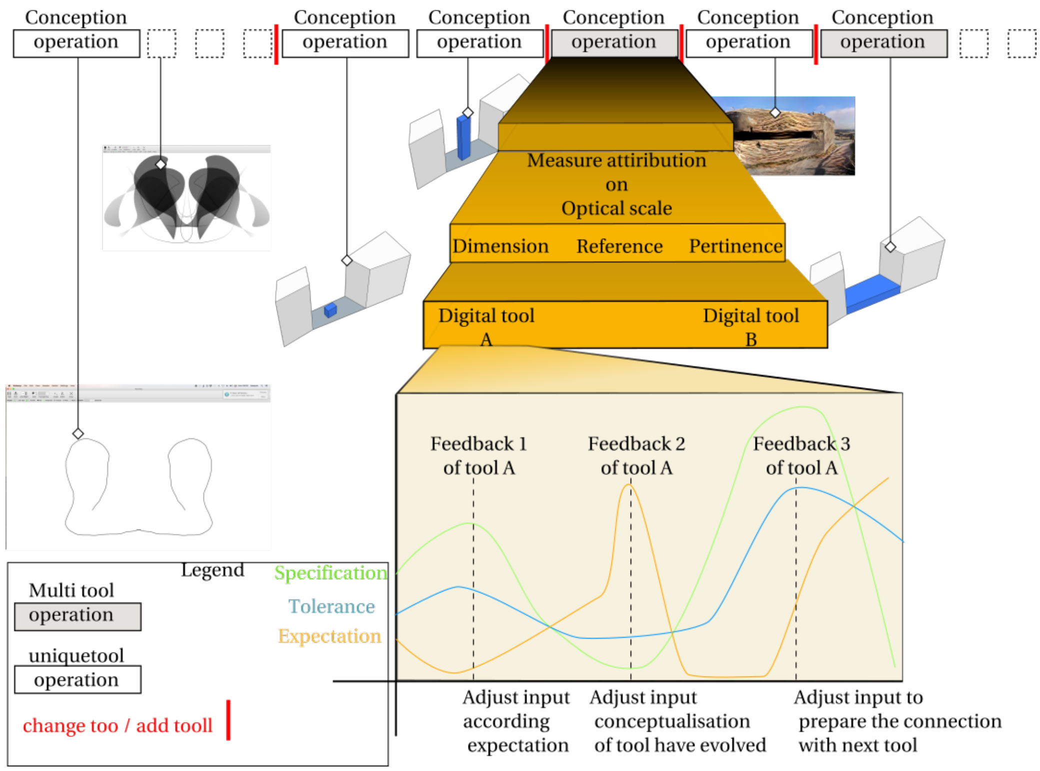


FIGURE 3.1: We are looking to determine the designer attitude when he uses a digital tool for designing

Based on the pattern produced with the comparative evaluation system, we figure that ratio of evaluation are related specifically to the User. By asking different designer to apply this evaluation system on set of 4 different tools we figure pattern in tool comparison but pattern exist between users too. Indeed, the value of evaluation are average of trend between users' evaluation but if user are compared through the pattern ratio discover in the average, we can see that regarding those ratios, their is cluster of users that gave the same ranking for the same tool in the same ratio. As sure that designers think differently, even on a very minimal operation such as using a digital tool, their evaluation of the situation may differ according their point of view. This mix between their evaluation and their approach of tool in general will be called : the design attitude. Measuring designer thought is a wicked problem itself since their thought are always changing and a test to decipher them may change them simultaneously. Face to this problem we had to decide a way to model and evaluate that can be possibly testable without using EEG or other physiologic measurement tools while being not too subjective. Since this time that's not the relation that is evaluated, it is not justified to evaluate "from the designers point of view".

Those parameters have been selected and named from the analysis of pattern ratio in the evaluation system, for instance the Filter is always the most open with ConvNet based tools and it seems clear that this feature is linked to the Understanding-Conceptualisation couple. Nonetheless, we've seen that user didn't succeed to grade those tools and that they had difficulties to find strategy create variation in the use. They quickly get their Filter and Flow criteria that drop after 5 attempts. On the other hand, designers that develop quickly strategies to create change on precise area and seems to develop an intuitive understanding of the tool kept a the satisfaction criteria above the others.

Regarding those clusters of designer's evaluation we came to think that a specific attitude of those designer may be related to the ratio between Filter and Understanding - Conceptualisation in one hand and the ratio between Expressivity and Understanding - Conceptualisation. Those ratio have been called Tolerance and Expectation. Then the Expressivity-Implicit ratio that is present in almost all participants can be related

As the Figure 3.2 shows, we propose a model that have 3 parameters to describe the user attitude : specificity, tolerance and expectation.

3.1.1 Specificity

The specificity is the precision and the amount of information that are provided by the user in order create the input message. This related to Expressivity obviously but in this experiment we look at how much the expressivity is used. The tool have a expressivity potential through the structure of input message that can be processed but it is up to the designer to take advantage of it. Depending of is understanding-conceptualisation of the implicit of the tool, the user may sometimes make the economy giving detail in the input and expect a correct "interpretation" of user intentions by the system. In fact, most of the systems don't interpret users' inputs. The case of ConvNet based tool

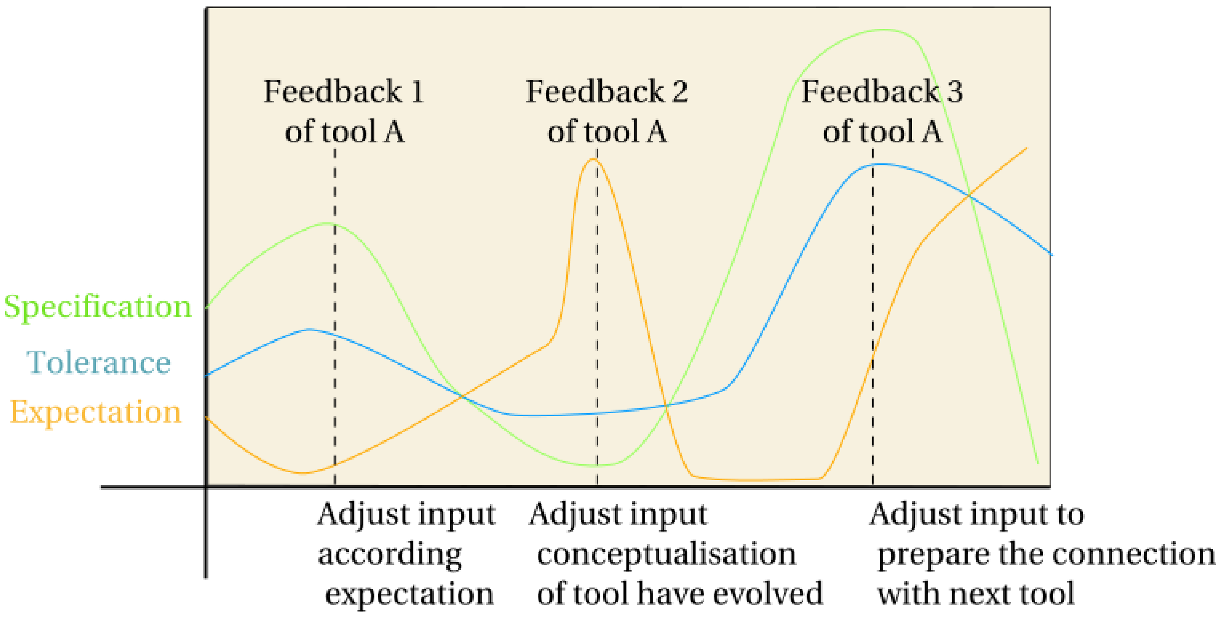


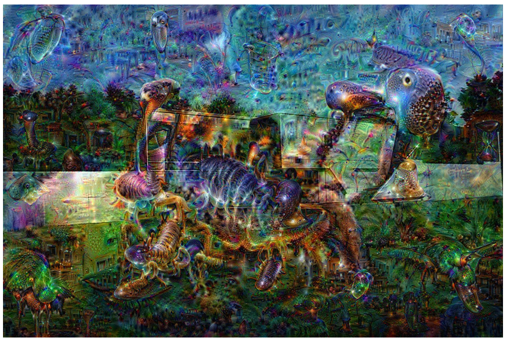
FIGURE 3.2: The possible fluctuation of designer attitude parameters

3.1.2 Expectation

The user expectations are what kind of result he anticipate from the input he provide to the system. Before the definition of the input specification, the user already has preconception of the tool that are part of his expectation. Those expectation may evolve through the feedback as seen before. Expectation influence three criteria of the previous evaluation system. Indeed, *filter*,*Understanding*, *Satisfaction* are decided upon the user's expectation. In the other hand this expectation is adjust on criteria such as *Expressivity*, *Implicitness*, *Conceptualisation*. This index that is measured here is average in unknown proportion of those different criteria and impact in unknown proportions post-feedback criteria.

3.1.3 Tolerance

Finally the Tolerance, keep in check the expectation in a range. Despite expectation can't be prefigured before the production, they may be more or less precise depending the mind set of the user. The tolerance may seem redundant with the expectation since the tolerance is somewhat the inverse of high expectation. In fact low expectation can't be assimilated to high tolerance since low expectation can come from a high or low specification but their is no tolerance regarding since the user consider the task as intrinsically doable or easily doable with so much specifications. Those parameters are linked but it is up to the personality of each designer to put weight on the "coefficient" that connect those parameters.



(A) DeepDream algorithm used on ConvNet made with GoogleNet Model trained on the ImageNet dataset

(B) DeepDream algorithm used on ConvNet made made with Places-CNN trained on the MIT Places 205 dataset

FIGURE 3.3: Depending the ConvNet use in Deep Dream the output present different "dream tendencies"

3.2 Deconvolution of ConvNet

As explained before, DeepDream is not initially made to be a design tool but it has been release to the public with the hope that the artist, designer and layman take ownership on it. The "out of the box" setting allow only to select a initial picture, a scale of patch, a octave level and a layer to "stimulate". Since it is not a software but a script that gather a library, a framework, a trained ConvNet and user inputs, lots of modifications can be done by the user if he has a correct conceptualisation of the tool. Compare to "simple" software with a WYSIWYG philosophy such as the *Add cube* function in Blender, there is more room of progression with tweaking of Deepdream script parameters. In the end, it is a programming interface so it opens to enable the design of any tool if the user have the right knowledge. That's why, the user's tweaking have to be limited in a range that don't change fundamentally the *identity* of the tool. The user have been briefed that additionally to the attended use: selection of a pictures and of few parameters, they can do any change on the image and change the code variable without touching loop or adding code.

Indeed, from the python interface provided by the Jupyter web frontend, experienced user can change completely the nature of the tool and invalidate the experiment. Despite we wanted to let them find hack of the tools in order to observe their creativity face to the tool, it was necessary to restrain the use option in order to keep the measurement of the specification in a countable range.

At the beginning a brief explanation of the tools parameters have been explained and in order to illustrate the the principle of the tool a switch of the ConvNet component have been shown. As Figure 3.3a and Figure 3.3b shows, the switch of the ConvNet orient the type of "dream" produced by the Deconvolution principle. The initial image provide a source of information that will reveal information encoded in the ConvNet. That's why, a same image submitted to different ConvNet generate different dream. Above that, it is important to notice that GoogleNet model is a deeper ConvNet – ie. it has more layers.

3.3 Test the model with Deep dream experiment

To test this model, we asked to designer to use the Deepdream deconvolution system as design tool. They had to do a digital operation that may take place in a imaginary design process. Deep Dream has been selected upon Neural style since the conceptualisation of the tool is more open to generate expectations. Nonetheless, with advanced users, we had case where a more precise conceptualisation of the tools led them to elaborate use strategies of the tools with higher expectation than the one initially conceivable.

The protocol of the experiment was to explain the basic of input manipulation that can be possible in order to have more control on the dream generation. Since the generation of image takes few minutes, we had time to interview the user on the expectation he had regarding the input message he produces. Concerning the specification, we look on the number of sub operation made to produce the images. Despite each operations doesn't bring the same amount of specification to the system, for the designer, they represent a specification effort that is not done gratuitously. This is a good indicator of the level of specification. Since we want to determine a model of the attitude, a general tendency is searched more than a exact amount of specification that is characteristic one design attitude or another.

Finally, to evaluate the tolerance, two factors are used, the debrief from the user after he saw the output. This output is judged regarding the expectation and the level of tolerance he self-evaluate during the deconvolution time. The reaction to the output help the designer to clearly separate his expectation from the tolerance. He can comeback on his evaluation of the tolerance made before seeing the output. Indeed, tolerance is partly expectation but it is mainly about the acceptance of unexpected result. Those especially that are the product of CR. That's why the user have to make two evaluations, one to know how much he think he is ready to accept as a satisfactory and those he indeed accept as satisfactory. Since it is a range, he need to know the two limits of this range.

The user that did the evaluation of tools have been re-contacted to perform this small design exercise while evaluating those attitude parameters face to the tool. Similar evaluation of attitude have been compared with the trend of their tools evaluation.

(A) If this picture is submited, can the tool make slight modification to this original model in order to get something similar to the pictureon the right ?

(B) This building is inspired from the Villa Savoy, nonetheless, can the Deconvolution system can generate something in the fashion of this building from the inital picture of the Villa Savoy

FIGURE 3.4: Having such trust on current ConvNet based tool capacity is over expecting but confronted to new kind of tools, designer need an adaptation time to identify actual limitation of the tool

The goal was to see if the evaluation of the attitude correspond to a model that may characterise one of the group delineate by the evaluation comparison.

The user made different attempt with the tool in order to see evolution of their attitude during the tool discovery. Indeed, the feedback provide them idea of what can be expected as output from the tool.

3.3.1 First expectation

After the explanation, before the first use of the tool, we ask the user what they expect from this kind of tool.

To define the level of expectation we can't judge expectation from the research point of view since his knowledge of the deconvolution principle becomes intuitively clearer after few experimentation of the tool. The Figure 3.4 show clearly a too high expectation regarding the capacity of ConvNet tools. Nonetheless, the user attitude can't be define from experimenter knowledge of the tool. That's why, in a pre-feedback time, the user has to define the level of is expectation without any help from experimenter.

3.3.2 Evolution of the Design Attitude through feedback

Other user like user B2 start with more pragmatic approach and had almost no expectation for the first attempt. He focuses on the setting of parameters rather than trying to manipulate the initial image. In comparison the user B1 directly try to control the area

FIGURE 3.5: With RGB noise, the user A try to focus the generation of dream on specific area of the picture

where the dream are generated. As we can see in Figure 3.5 B1 tries to generate things in the dark area of the picture. The area are judiciously selected in order to increase the chance that the dream will fit to a local context of the picture. Indeed, previous use of the tools showed him that the dream happens sometimes with context but empty area are filled with dream despite the absence of context. Indeed, nothing in the algorithm is plan to inhibit the dream in certain areas. The original tool was design to test the ConvNet, since the initial image is just used as pretext to unveil tendencies of the ConvNet layer, there is no such mechanism in the tool. When the dream seems to be related to a context of the pictures, its indeed based on pictures local features but if nothing is found, rainbow artefact is generated. From this feedback, the B2 user's expectation get higher and he submitted a picture with more specification. He wanted to use the system as pattern propagation system. According to his vision of the tool, the tolerance was pretty low since he was sure of the result of his input.

3.3.3 Attempts to orient the dream

In this section, we will show different strategies that are developed by user in order to use the tool for their design goals. Those "strategies" are the source of information used to calculate the specification. For instance, we can see in Figure 3.6 the propagation of the branch tree pattern in the sky. On the basis of this effect, application for design representation can be imagined. Propagating a texture on a surface automatically and with a natural aspect can be a valuable tool to foresee the look of a building with

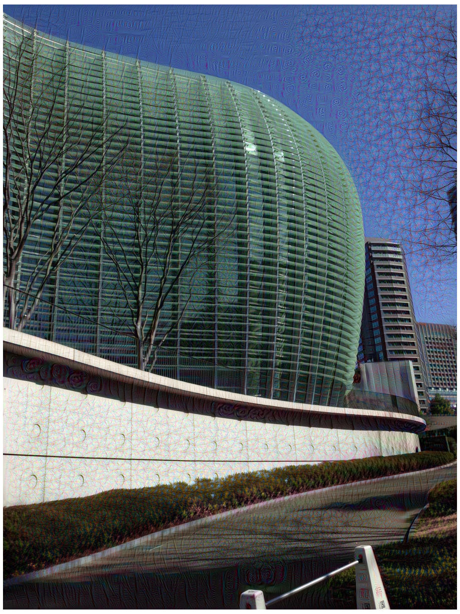


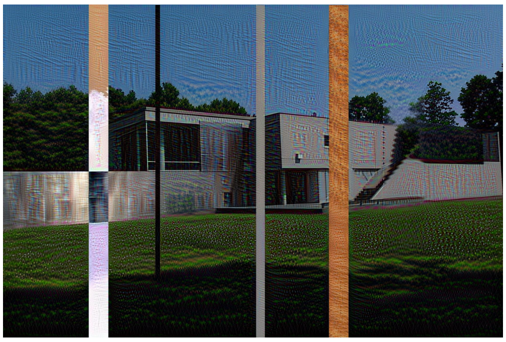
Figure 3.6: We can notice on the right side of the picture that the pattern of tree branches is propagated in the sky

creeper on it. It could be initiated by a easy photo-montage like in Figure 3.7 and then the DeepDream will propagate it if it activates the appropriate area. An other designer

Figure 3.7: With this specification the user was expecting to have a kind of creeper on the wall of the house

tried to trigger propagation of pattern in a similar way. As we see in the Figure 3.8, the Figure 3.8b doesn't display any propagation that "start from" the texture band put on the original picture. On this case, the designer said having low expectation. Just

(B) Then the user get more tolerance and lower expectation and enter in a more experimental phase in order to understand how the tool behave



(B) Despite very low expectation and a high tolerance to the result, the User B2 was decieved by the result

FIGURE 3.8: The different texture supposed to activate some neurons of the ConvNet didn't meet Designer's expectation

User	Specification	Expectation	Tolerance
A1	3	low	high
A2	4	high	high
A3	6	low	high
B1	1	high	low
B2	3	low	low
B3	2	low	low

FIGURE 3.9: Designers estimate their Expectation and Tolerance on a very simple scale

expecting some propagation without being conceal in a meaningful area of the picture such as a portion of a wall.

The goal of the experiment was to test expectation and not the satisfaction of the user. Nonetheless, the User B3 gave up on using DeepDream as design tool. And in concordance with the Tool-User relation model (Figure 2.7), the parameters of the designer model seems to evolve with time through the Understanding criterion. Since the evaluation of tools has been made globally by set of 4 tools. Understanding evolve and the comparative evaluation change after few attempt. Nonetheless, the dynamic of the ratio is changing in a predictable fashion. Indeed, even if the *filter* tend to change for a close type, in comparison with very different tool, the evaluation is still valid after few attempt. That's why we finally made an average of the 2 and 5 attempt for each use in order to get a characterisation of their attitude. we can see the result of this evaluation in the Figure 3.9. The high specification seems to put the designer being less "tolerant" about the unexpected or randomness of images. Indeed, the designer that had low specification, didn't make much or complicated modification of the input picture, were more expecting for inspiring product than final one. Thus, they got less deception since their investment in image production was less intense.

Finally, comparing the group that had the same type of evaluation of tools and their design attitude evaluation seems to show that group that emerge in the two evaluation seems to match. Despite the low number of participant, we can define an designer attitude that we call "open design attitude". As shown in the Figure 3.10 We characterize it as High or low expectation but with high tolerance and few number of specification.

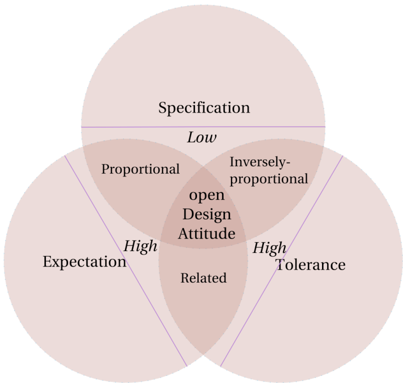


FIGURE 3.10: Designer attitude need to meet particular condition in order to be receptive to resonating feedback of the tool

Chapter 4

Flow in digital interaction

4.1 Measure of the Flow

In psychology, the state of Flow have been studied by Jeanne Nakamura, Mihaly Csíkszentmihályi[70] and can be described as an intense focus during an activity. Not specifically related to a "flow of interaction" with another entity, this concept still present connections with the concept used in this thesis. Indeed, according to this theory, the mind can process a limited amount of information per second and so, in a similar way, the feedback informations provided by the tool may overflow the mind capacity. In the same vein, flow theorist advocate for the importance of "immediate feedback" and on this point specifically the study follow their approach since one of the variable that will be modified during the test is the feedback speed. On the question of matching skills and challenge, the experiment presuppose that individuals who have performed the task of the experience are designer. So, as designer, they intuitively perform an act of problem finding during the exploration of the tool. They set achievable goals adapted to their skills and tries to fulfil them. Concerning the balance of skill and challenge, it is partially minimised through the shortness of operation allowed by the tool. Indeed, short operation allow rapid feedback and quick fix of the tool use micro failure. Especially with Fluxus that allows quick compile test at various step of the code writing or Al.chemy that push the users to pass over their feeling of failure by being a palimpsest for constructive perception. In the difference to Fluxus, frustration can arrive if the user don't succeed to construct a perception while Fluxus can generate it from the recurrent compilation failure. Nonetheless, they were the two best tools to test the Flow in a seek of CR research since they are tool based on short input, rapid feedback and oriented for an open design attitude. Where the concept of Flow in this experience differs from the original literature is on the importance of a preconception of their goals and the consideration of antecedents of flow is just taken in consideration through the experience of user. Above that, in order to observe flow in creative task that are conducted in attitude comparable to an *open design attitude* we decided to don't provide a specific goal for the task. An important point of the literature about Flow is it's measurement. Three types of Flow measure exist, Flow Questionnaire, Experience Sampling Method and Standardized scales. In addition to that, psychophysiological measure can be taken such as pulse, arterial pressure, facial muscle, ect.[71]. We focused on indicator of flow and orient a flow questionnaire in such fashion and put the balance of challenge and skill as a component of flow as the model of Jackson and Eklund [72] propose while keeping a form close to the FQ (Flow questionnair) since it This questionnaire have been given to

4.2 Measure of the Flow in design task with digital tools

As seen in the digital tool evaluation system, the flow is partly the time of feedback response. This duration is mainly due to the hardware and / or the code optimisation. This evaluation seems at first sight not related to the user nor it's relation to the tool. If we look in depth, the situation is far more complex. As the Belgrano Workshop survey and observations uncover, the lack of rapid feed back between step and the length of the process made the use of this tool chain hard to master at the first use. Above that, the inherent length of the process was clearly suspected of being the cause of lots of difficulties for the group that fail in the construction of their model. In the CAADRIA Workshop, the participant were free to build their own tool and the software teach for creating this tool was based on a rapid feedback philosophy. Indeed, Fluxus is designed for live coding so the output is always present in the background and the programming language is made in such way that it promotes operation on data stream instead of state as it is usual in most of the popular programming language used nowadays. Despite those rapid feedback provided by the IDE, users were not achieving any Flow state in their practice of the tool because they were beginners with the tools. If live coding session preformed with Fluxus are observed on internet[1], we can see the Flow potential of such tool. But such flow is staged for performance and doesn't relate a design process that will not conform to similar requirement of performance. Even if both are not incompatible, the aim of Flow in design practice based on digital tool is different than live codding. Fast feedback are always welcome but since the output need to be well observed in order to decide the next design operation, the flow of design can be a bit slower than the performance pace. The pace for design practice is aiming to avoid the change blindness that human brain is subject to. Indeed, with computer tool, the speed of output or animation is set on the clock speed of the computer and it may happen that result or

[1] https://www.youtube.com/watch?v=XF0taJpAesQ

animations are too fast for user's perception. The change blindness, happens for very little, or very slow modification. For a performance, to rapid change is not a big problem if other change are slow enough to captivate the audience attention.

4.3 Efficient use of the hardware

As said before, the speed of the output, even if it is not the only parameters has an important role to play in the reach of a flow state. A very common digital tool experience is to experience the computer that lag when a complex operation with lots of item on the screen is performed. The user loose his focus on the operation and shift on the technical problem that is the cause of this lag. All slow down of the computer response don't trigger a loss of attention of the user but it opens the opportunity to wait for the output. During the waiting time, the attention of the user may drop because the changes are not perceived in pace that is similar to "non digital feedback" provided by physical tools for instance. The most common lack is the lag, blindness of change on too fast modification may happen but usually, it is mainly due to a slowdown of the system. When designer create their own tools by programming, a missuses of the memory management or a number of object state per frame to compute to high can create those type of unpleasant experience for the user.

4.4 Chain of tools, design a representation production process

As said before, the flow in digital tool use for design doesn't imply a speed of use similar to live codding session and it doesn't even imply use of code at all. Indeed, it really depends of the skills, habit and comfort with the use of tools. For instance, the User A2 in the previous experiment shows great ability in the use of Al.chemy and GIMP. Despite the small time of transition from one environment to another through a file transfer operation, the user have a tool chain that is, regarding his skills and habits, efficient for let him achieve a flow state. Only frequent and long waiting time of more than 20s seems to disrupt the design flow for this user. Since the flow seems to be related to experience, we had to ask to experienced user to perform test for us. Indeed, when designer such as Kostas, develop a tool that may have a CR potential in their practice, the development of the tool and the practice are probably done in a state of Flow. We can imagine that somebody who spent time on the development of *Form Z* have probably abilities in code and those abilities in code should be taken in consideration as reason of achieving CR in this computational practice of design.

Figure 4.1: User A1 had a initial intent but it progressively drifted to another draw

4.4.1 Necessary but unrelated to design: data management operation

Unfortunately, use of digital tools is often an alternation of tool technical problem and design problem. Indeed, some operations performed on the computer are not related to design solving problem. They are call usually technical problem and are often about software installation, software or OS crash, file import and export, version management... In realist study, they can't be ignored because we assume that designers know the difference on this particular issue and will not mix-up their design though with those contingent problems and task. This can't be ignored especially with tools that involve the connection of tools in API or libraries through scripting or programming language. Indeed, some design of digital process stay unachieved due to those problems and discontent. They could be classified as inherent possible feedback of the digital tools and count as a design feedback. A type of feedback that express to the user the invalidity of the tool chain implementation, and so partly its design. For design of house for instance, the time either valid or invalid the designers proposition. Such rapid feedback is a vector of quick learning that is specific to codding practice, and technology in general. Ultimate criterion of legitimisation is: is it working or not. The satisfaction of a working process can hide the vacuity of the digital tool by product or worst, make believe in liability of the output higher than the real value. Sherly Turkle in her book Simulation and its discontent[73]. That's why usually, designer prefer the comfort of designed environment and staying in the range of the tool kit proposed by the software ensure trust and avoidance of digital tool binding that doesn't work. In this research, we take the position of putting the designers responsible of their tools without assuming that they should know everything of the tool and its component. It is a shared responsibility but since there is always disappointment, the designer need to think forward if he want to develop effective design strategies that involve digital tools.

4.4.2 Parametric substrate model

A parameter that will be determining in the construction of a digital design strategy is the format of the substrate model. Indeed, digital designers don't create and / or use tool pure data as a digital media designer will do. Indeed, they are bounded to the architecture context and their final product is expressed through the construction of human scale object. For this reason, there is a substrate model that receive and gather in the end all the conception operations performed in the design strategies. This model is usually a 3D model that can be enriched with non geometric data bounded to parts of the model as it is done in the BIM format. Sometimes, the final substrate model can be different than the working model. Indeed, for the Belgrano Workshop, the final substrate model is the physical model in wood but the model on which design operation is in between the various representation in the 3 sub-tools used in this process. To avoid the unrelated to design digital operation, designers tries to find digital tools to bridge automatically their data to one 3D modelling software for getting visual feed back and ability to make on the fly modification with the tool-kit provided in the software. In order to improve the incompressible manipulation time and enhance the flow in the design practice process, the substrate model format and the unity in term of access is key part in the digital design strategy. For the measure of the flow in the digital design, we will not take in consideration the tool design, setting or modification. Even if according radical algorithmic designers, the design is the design of the process itself, we don't have the conceptual tool to measure a process that can span indefinitely: a full tool design process. That's why we will focus on the sole use of this tool without consideration for its design and implementation part.

4.4.3 From one modification to the feedback

The measure of the importance of flow for CR is difficult to put in evidence since there is no perfect case of tools that ensure CR with all user in all situation. ConvNet tools seems to have a valuable potential but the available "tools" are hack of research tool and have no been optimised for design process. AS we can see in the chapter 3 and chapter 5 they can be used to get indirect measurement of parameters important for CR but in the case of Flow measurement, the available ConvNet based tools don't provide option for testing different pace of interaction. Indeed, deconvolution takes about 5 to 10 mn on a computer with i7 chip and NVidia GTX980 graphic card. Nowadays, it is a powerful and expensive computer configuration and it is not possible to get better at reasonable price. No faster computer is available to test if a shorter feedback may create a better interaction between user and tool. To understand the importance of the output generation duration and the input construction duration, only observation of other tool with no CR feature can help. Indeed, tools that have rapid feedback possibilities such as Fluxus can get their feedback artificially elongated. On the other hand, the input message construction duration depends on *Expressivity*, *Implicitness*, *Conceptualisation* and other criteria of the tool-user relation. For this reason it is really user dependant and more precisely, user's experience dependent. Even more than the experiment with neural style, the user experience is a variable hard to control. Indeed, since *Neural style* and *DeepDream* are pretty new and unusual tools for design, most of the designer that perform the experience had the same experience level. Even if their personality differs, their knowledge of the tool working was basically similar. In this case, since we can't use ConvNet based tools for the experiment, two tools will be used for measuring separately the two parameters of the Flow. Even if this Flow is not in a case of CR, we can proceed to an extrapolation form the non-CR potential tool used in this experiment. A level of flow in evidence in the experiment this chapter combined with the *open design attitude* of the previous chapter and the *context encoding* explained in the next chapter can give insight of what can be CR experience.

To study flow, two tools are used : Fluxus and Al.chemy. On both tools, the variation of the duration of the feedback is possible since the output is by default instantaneous. Nonetheless, both tools have one parameters that is slightly different: Al.chemy is pretty intuitive and the input message generation is very natural for people who practice drawings with real pen or on graphic tablet while Fluxus, despite all the effort made to propose a programming environment for rapid interaction in code design, the construction of input message is intricately longer and more complex due to the non-intuitive aspect of programming for average designers. Since both tools, accumulate operation effect on a "substrate model" of a format suitable for the actions of the tool-kit provided by the environment. We measure the time of input creation and the output generation time. Since it is the controlled variable, we selected three delay duration. Those durations have been selected on the basis of previous literature about VR controller [74] for their short values (500ms 1s) and the long value had been decided based on literature about remote control of mechanical system. Nonetheless, we could not follow those previous methodologies since they are presupposing the realisation of a task. Indeed, in a task completion context, the lag, according to these papers, is degrading the performance. But for design process, time can be an ally since it creates a span of time for thinking. Depending on the tool used for design, this lag time may not be a problem.

To conclude the set of the lag is set from the end of the input message construction to the related feedback. Then in order to have an insight of the input construction duration for experienced user and novice user, we indicate an average range of construction input duration. Al.chemy and Fluxus present different working and the construction of input is asynchronous for Fluxus and synchronous for Al.chemy. It follows that a measurement method should take in consideration this parameter. To face this situation we take in consideration the duration between each stop or take of the page is taken as time of the input construction. For Fluxus, from the output to the

4.5 Automation of the chain and instant feedback

Fluxus is inherently a tool to construct a chain of tools and its asynchronous feedback makes it essentially different than Al.chemy. Even to trigger the interpretation of the code, the user need to type a key binding. On the other hand, Al.chemy is not about creating mediated mean to produce an output that will be use for design. The output is a goal in itself and its correlation to input is slightly deviated through the effects that can be selected by the user. Measuring the feedback of this two extreme case present the interest of comparing two tools that despite presenting no encoding of any reality context, are tools oriented for *Open design Attitude*. Indeed, Fluxus like Al.chemy, promotes in their design and their philosophy the flow, the high tolerance to errors and a progression by short chunk of *specification*. Fluxus program borrow his name to an art movement of 1960 that promote a fusion of art and life and aim to intersect media, experiment with a lightly state of mind more curious about the outcomes than reaching a level of quality. Thus it seems very compatible with the *Open design attitude* described in the previous chapter. On the other hand, Al.chemy is presented as "an alternative way of drawing" and advocates for unusual software design choice such as the absence of the very common "Undo" command, or the lack of editing and selecting command.

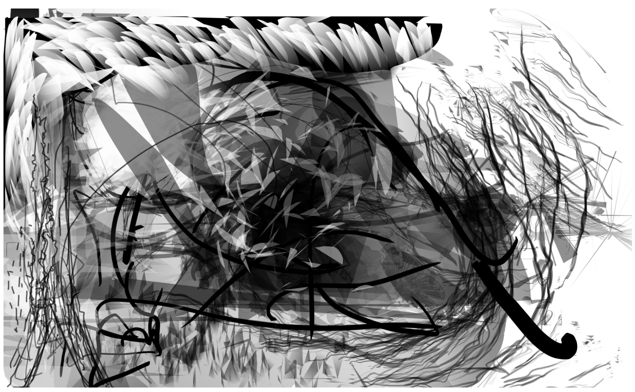


FIGURE 4.2: User B3 have try to use Al.chemy with the same Philosophy it had been designed.

Nonetheless, the "Randomise" command modifying the current substrate model so it can be seen as a edit command. Otherwise, every other tool mechanism is based on the addition an accumulation until the user discover through is constructive perception a lead for inspiration and export the sketch in another, more classical vector or raster graphic editor such as GIMP or InkScape.

4.5.1 Toolkit environment

As previously mentioned, full software is often erroneously referred as a tool when in fact it's a framework and its corresponding tool-kit. Those two "tools" don't escape to this definition and the input construction and the output generation are not always measured between the same tool but between tools of the tool-kit. It may happens, for Al.chemy that more than one time in a row a same tool may be used but it doesn't change the problem of flow measurement, tools are often part of a software environment and the flow of repetitive use of the tool itself depends of the nature of the tool and there is no "silver bullet" tool that is sufficient to conduct a full digital design in a computational thinking approach. If a stand alone tool, is proven to have a CR potential, the question of is integration in software environment or its fluid linkage with other digital tool is a point to take in consideration for assessing of its CR potential. Tools that present, sequential and iterative use are tools that can achieve more easily the flow experience. Nonetheless, those tools are usually not really fitting with a computational thinking approach or with A.I. features. Nonetheless, we can imagine tool with CR potential, based on Computational thinking and being the fruitful in his creation of a process design. For instance, a tool that interpret free hand draw in photo realistic texture and shapes. Based on ConvNet trained on hand draw this tool may recognize very naive drawing and replace them by an accurate delineated part of a photography that represent that tool. In this case, the Flow of a CR tool could be perfectly measured.

4.5.2 Programming time and runtime

For the Fluxus case, one specificity is the mix between the programming time and the runtime. For this reason, the tool is close to this iterative sequential reuse of the same tool. In fact, it is the reuse of a tool that is changing after each use. Despite the evolution is not propelled by an AI force, it is still in CR philosophy since the oscillation between user and tool may lead to *a greater amplitude* of effect on the project and designer mind through the accumulation of "intellectual" energy like a oscillating system may accumulate vibrational energy. In this case, its exclusively the user Intellectual energy. This gradual accumulation of "intelligence" in the tool show its value only on few last interactions when the tool-user couple get the most fruitful collaboration. In this moment, we can talk about CR and that probably that kind of moment in the design process which are reached by designer when they succeed to create a system that is mirroring their idea of a computational design process. That was probably the case when Kostas Terzidis get the first significant result from its permutation design program. In a similar way, an experienced programmer and designer can achieve the same kind of performance with Fluxus. It will be hard to prove one day that the flow of fluxus environment had improved the chance to reach this CR level but at least, with the next experiment we can show that Flow depends of Rapid feedback, short and iterative input operation and intuitive interaction. Concerning the "intuitive" aspect, we will see with experienced user of tools that intuitiveness of the tool can be understand as experience and skill of the user. As stated form the beginning, CR is as much dependent from the tool than from the user.

4.5.3 Design process as a sequence of digital operation

The design process studied here is the one that stays within the boundaries of digital operation. Mental operations that have no concrete manifestation that can be recoded or detected are ignored despite they present a valuable interest in the design practice. In a pragmatic attitude, we focus on phenomenon that have physical manifestation in order to elaborate theories on unseen ones. For this reason, the test have been limited to see if intuition, rapid feedback and short input messages are a source of Flow. To see if the 3 are necessary, we took indirect factors to control the variation of the parameters :

- intuitiveness is related to the experienced and inexperienced variation parameter.
- Both software encourage short input but Al.chemy is naturally more intuitive since it is partially designed to be a metaphor of the drawing table
- the speed of feedback is the only parameters that can be controlled precisely and directly by changing a small part of the initial code of those open source software.

The definition of a state of Flow in the interaction is defined by the user through the help of the adapted Flow Questionnaire and can be confirmed by their input construction time for the short input delay case. Initially, these value was recorded without a clear objective, the interview of users were supposed to be enough, but for each tool, we note that above a certain duration to produce the input, it was sign, even for experienced user, of slow down of interaction and thus lost of Flow.

The experience lasted 5mn for each session of Al.chemy with different feedback delay setting. For inexperienced user, 15mn were used to warm up on the tool. Concerning Fluxus that require more minimal practice time to start to be autonomous, a short class

User	tool	Normal use	500 ms	1s	2s	input
user E-A	Al.Chemy	Flow	Flow	disturb	disturb	0 - 2s
user E-C	Al.Chemy	smooth	smooth	disturb	disturb	0 - 2s
user I-D	Al.Chemy	smooth	disturb	disturb	disturb	0 - 5s
user I-E	Al.Chemy	rough	disturb	disturb	disturb	1s - 5s
user E-D	Fluxus	smooth	smooth	smooth	disturb	500ms - 2s
user E-F	Fluxus	smooth	smooth	smooth	smooth	2s - 20s
user I-B	Fluxus	rough	rough	rough	rough	10s - 120s
user I-A	Fluxus	rough	rough	rough	disturb	60s - 3mn

FIGURE 4.3: Evaluation by user of the flow and smoothness of their interaction

of 30mn have been provided to inexperienced users. Then, 20 mn of play-around for warm to finally spend 20mn to "design" for the experience.

As the table shows, the 3 parameters are important to reach a flow state. Indeed, in the Al.Chemy case that present an intuitive interface and short operation, the delay is more disturbing for user and especially for Experienced users. Those user can be considered has having the highest intuition level since their experience and the nature of the tool concur to increase this factor. For them, the delay was always disturbing while Fluxus users, that by nature of the tool need more time to create input since it depends on typewriting skills, 3D space orientation and habits to functional programming. For them, operation are longer and change may be harder to notice since nothing indicate that code interpretation failed. For them, the feedback may be a problem more because it breaks their habits than clearly create a pace unsuitable for design. As the E-A2 user did, longer delay in the feedback was used to think about the next operation. For inexpericed user, the slower feedback doesn't change that much since initially they need long time to create input and understand what is wrong in their code. Nonetheless, inexperienced user use a lot the compilation to check if their code is ok. The delay in the feedback was annoying since even in their struggle to use the tool, they found a pace of interaction that allow them to learn quickly from their errors. Delaying the feedback break this error based learning flow. For this reason, even if intuitiveness through practice experience seems to be a predominant value in flow reaching ability, we can argue that even with a lack of knowledge and experience flow can be achieved through the learning curve of the software. We can presume that the preponderance of intuition-experience is over rated and mask other aspect that can create intuitiveness in the interaction. The GUI design, the scarcity of tool and the existence of clear metaphor to understand them seem as important than experience.

4.6 conclusion

As conclusion, we can say that Flow in the use of digital tools is about a balance of those 3 parameters more than a precise threshold value that need to be reached to get the flow interaction. It's depending to user but since it's a balance, the speed of feedback is a parameter that can be adapted according the intuitiveness of the context for the user. This feedback can be adapted to a slower speed only if the computational power required by the tools' algorithms are below the hardware capacity for real-time rendering. By contrast, the length of operation is a parameters that is hardly modifiable on ready made tools. It's directly part of the tool's design and need to be decided upstream in the design process of the tool. It is directly a part of the philosophy and use context of the tool. Photogrammetry software are not working in real time because they proceed a clear operation and the time to create the input message is not a problem since it is use to create a initial substrate model and not to work on a model in interactive fashion. When IA or algorithms are well enough designed to solve tamed problem, intervention of the user/ designer is not necessary. Indeed, all the work of decision is already staked by heuristics or scientific rules.

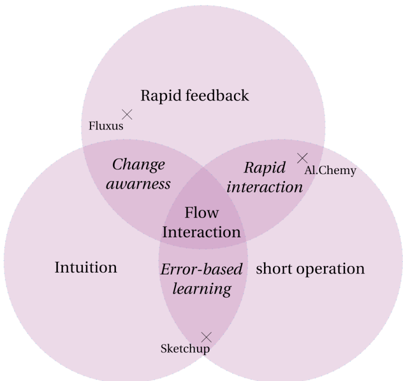


Figure 4.4: The oscillation of this communication is possible only if the input and feedback pace meet brain capacities

Chapter 5

Measure Implicitness for Computational Resonance

5.1 Neural Style experiment

This third experiment aim to give a measurement of the parameters that can explain the slightly strong ratio of implicitness regarding the other criteria of the Production Orthogonal View (POV). Indeed, this criterion seems prominent comparing to other and usually, other tools seems to have more harmonious values in the POV. Those value are more balanced in WYSIWYG tools and unbalance is specific to programming tools that have theoretically unlimited expressiveness. As seen in the previous chapter 4, there is a notion of *intution* that play a role in the flow of interactions. Nonetheless, various parameters have been proposed to explain what are the causes of the intuitiveness of such tools.

The two tools used for the experiment were chosen because the presumably closest tools to CR were lacking of possibilities for adjustment of the feedback response time. To find a measurement of the parameters that explain higher implicitness rating the notion that will be tested through this experiment is the Context Encoding (CE). By this term we mean information about the context that are embedded in the tool through the form of computable data. This context is, for the digital tool, the mind context of its user and the physical context of the *matter* on which the tool is working. In the case of architecture design it can be the site or environmental data. But more generally, it is the perception of those data by the user. What they mean and how he understand them is the real context of the tool utilisation. The primary context is the designer mind and in this context, a sub-context is the idealistic understanding of the physical context by the user. We use "*idealistic* understanding" because we implicate an always truncated understanding of the site context that is always taken as complete understanding. Indeed, in design, the data of the site used to take decision can be understood in a sensitive way as much as a scientific way.

Through this experiment, we are going to show an indirect measurement of the CR through the CE that exist in ConvNet Based tools. This CE is expressed through the implicitness of the tool. This notion seen in the chapter 2 is the part of information that are implicit in the input message. In most of the tool, the implicitness is the context of the tool itself. The user know implicitly that it is not necessary to precise in its input message that the context of the input message is a 3D Cartesian world since it is concomitant with the use of a 3D modelling software. Rare are the tools that have implicit information hard-coded in the tool. We can have example of such implicit information encoding in IA of personal assistant such as Cortana, SIRI and M from Facebook. This tools have their use implicit context but added to it, their is other implicit information. Since we don't *need* to know those implicit information to use the tool, we can experiment how much implicit expression of the user overlap implicit embedded knowledge of the tool.

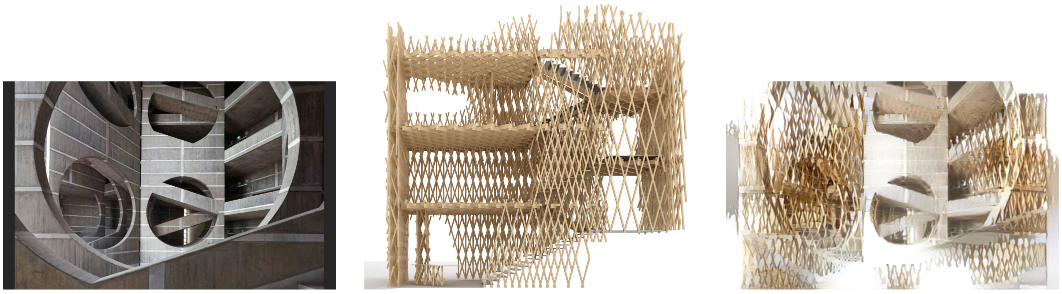
As seen in the chapter 1 he use of digital tools in the seek of CR goes beyond the conventional idea of tool utilisation: tools are just tools and only count the will of the user that is in full control of the outcomes of his tools through his input. The place of the tool in CR use of digital tool for design is different from this usual idea since the tool influence is seek for open the user to new horizons. For this reason, the designer that follow such practice approach doesn't aim to a full control. He looks for valid proposition while not giving all those specification of what is a valid one. Like the *permutation design* shows, a range of correct output is sought. Not one precisely. In this range the designer take the licence to make an instinctive choice of bathroom configuration by knowing that on a technical aspect, he can't be wrong. Similarly, in this experiment, we are going to measure implicitness by looking how much of input data are necessary in order to get a solution in the range of the desired one. In order to have a sample of the desired one, we asked to the user to create with other tools what he was expecting as output regarding the input provided to the system. By comparing the difference between the output of neural style and the expected result image we get a percentage of similarity. This percentage of similarity is varying regarding the number of precision provided to the system.
The experiment presented here is about observing the interaction with a system that contains AI components. This component is working with machine learning principles and is used to generate pictures with artistic style. By providing a style as a painter's artwork and content as a photographic one, the Neural Style [62] program creates a picture depicting the content of the photographic image with the artistic style of the style image features.

For this experiment, the implementation of Neural Style paper has been used as an architectural design tool. Originally this program has been made as a model of artistic style human perception. Later, through its Torch implementation [75] and release to the public it becomes, like the Deep Dream program [61], a tool for image creation. Indeed few designers and artists owned this tool in order to explore this new way of creating images. We use this tool for the experiment because tools with AI principles are seldom and this tool can produce images without much knowledge about Machine Learning from the user. Since the tool communication input and output are image-based, the tool can be easily integrated in an image production process of a designer. Indeed, unlike programming, the designers can use only a mere image to control the tool. At the core of Neural Style program there is a ConvNet. This ConvNet is the AI part of the tool. Indeed, all the rest of the code is usual data processing algorithms in order to take the advantage of the trained ConvNet in order to generate images from a system initially made for image recognition.

5.1.0.1 Other tools based on Deconvolution

The principle used to generate images with a ConvNet is called "deconvolution". Other programs based on the same principle exist. Indeed, systems like Deep Dream seen in the previous section, could be a candidate for this experiment since they process and generate images. Originally, Deep Dream takes as input only one image and reinforces features of the image that activates a user specified neuron layer. This system can be used in a similar fashion as Neural Style with two pictures in order to create a sort of blend of them. Both could be used but the discovery of an architectural design application of Neural Style makes it a suitable candidate for the experiment. Indeed, those systems are not originally designed as design tools. In their normal use, they are experimentation systems and their output are too rough and subjectively artistic to be framed in an observable design exercise.

Other systems like Deep Convolutional Generative Adversarial Networks (DCGAN) [76] can propose interesting interaction experiences. With this system, designer can perform a sort of arithmetic with images. For instance, to create a picture of a woman with glass, a "ConvNet representation" of man without glass is subtracted from another one of man with glasses then added to a last one of a woman without glasses. The "ConvNet representation" can be imagined as a map of neurons that are activated in the ConvNet when a specific type of picture (i.e. the pictures of man with glass) is used as input of the system. Such system can be used in architectural design for creating artificial images of architecture by providing various images to subtract and add.



(A) Despite the user wanted to use this picture as a content, the style of Louis Khan can't be taken apart form a content

(B) This image can sumarise the style of Kengo Kuma during a relevant time of his architectural practice

(C) Even if it was a good surprise, this result was not satisfactory for the user.

FIGURE 5.1: "A naive attempt to apply neural style algorithm to architecture field"

Beyond the technical aspect of using ConvNet at the core of their system, those programs have another common denominator, they retrieve from a "past experience" (their training) encoded information about characteristics of picture's elements. That information is at such discrete level that it is difficult – even for their creator – to understand in detail the behaviour conducted during deconvolution process [21]. This past experience is a sort of individuation process [50] of the ConvNet. The matching between this individuation and the designer one may strongly "move emotionally or intellectually" the designer. Indeed, despite designers know being in interaction with a machine he or she may unconsciously attribute this "resonance" feeling to the tool's ability despite the fact that it is mainly due to the partial overlapping of designer's and ConvNet training context. The test to put CR in evidence in this paper differs from other AI test [43], [46] where the testing idea implies to obfuscate the production method of the output. We propose the Computational Resonance term to describe such phenomenon occurring while interacting with ConvNet based tools because AI measured in tools is relative to the designer's "natural frequency" instead of the machine's autonomous trickery ability.

5.1.0.2 Unsatisfactory outputs

Before finding that restricted use of neural style that can be measurable, we get numerous attempt that have been initially created for the open design attitude experiment of the chapter 3. Those pictures are relevant to show the over expectation of some user's design attitude but they are shown in this chapter since they are related to Neural Style tool and that they show how over expectations were focused on how the context was encoded in the tool. For instance, in Figure 5.1, we can diagnostic that the user was expecting the notion of transparency was somewhat encoded in the ConvNet. One of the user/ tester is an professional illustrator and he was curious to generate draw in a manga style. Despite it was very close to the original purpose of the tool, his attempts show spectacular failure in the sense that the final output were, whether impenetrable whether created a "new artistic style" very disappointing since it was very far from initial expectations.

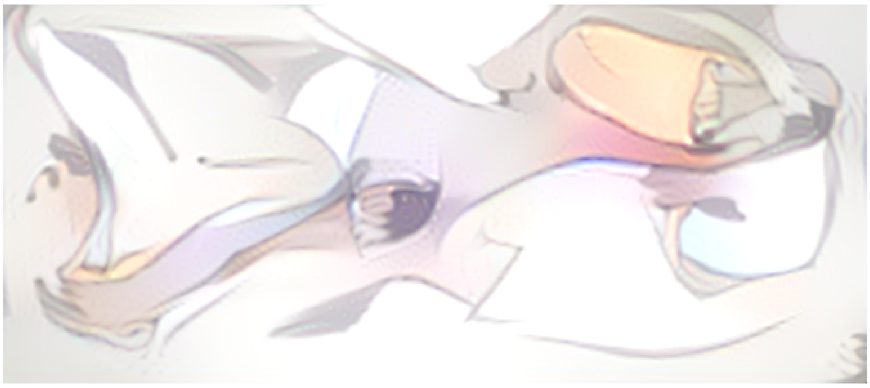


FIGURE 5.2: Close to a random image generation, this output was not considered as valuable by the user

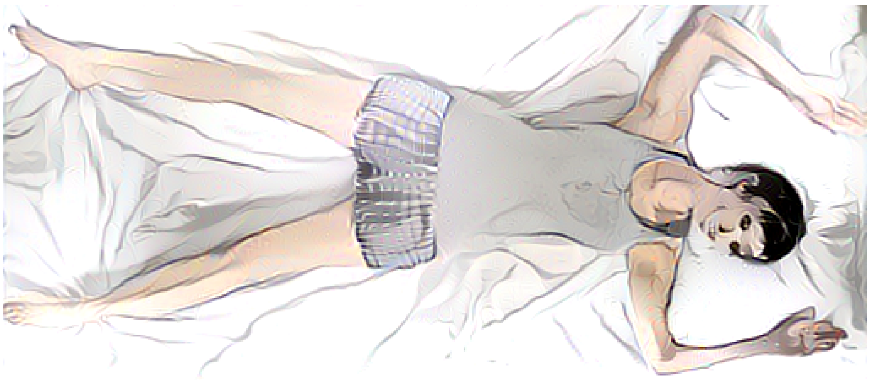


FIGURE 5.3: This style was not expected and it doesn't have any qualities that match with initial specification

5.1.1 Neural style algorithm process

Originally the Neural Style algorithm has been created as a model of the human perception and creation of artistic images. The original use of the algorithm is to provide input such as:

- on content image that should be a photographic image of the scene the user want to represent in the output
- a style image that indicates with which style the user want to represent the content image

(B) This content will be represented with the style of the image B

(C) This impressionist Van Gogh style will be applied on the content image A

FIGURE 5.4: This tow pictures are the attended inputs for the Neural Style Algorithm

In output it provide an image of the content A represented with the artistic style B.

Beside those basic setting of the input, other setting can be adjusted.

- setting of the number of iterations
- style and content weights
- Total variation weight
- initialisation mode
- link to the ConvNet file, thus selection of ConvNet file

Similarly to Deep dream, the algorithm increase the activations of specific neurons in the network by modifying the image. The difference here is the possibility to activate specific neurons of the layer through the style information pictures. To reinforce the modification of the initial image, like with deep dream, their is iteration process that reinforce the mapping of the style on the content. As show Figure 5.10, we can see how progressively the image evolve to homoeostatic state through the iterations. As we see, if the user dive in the options, lots of setting and experiments are possible. Since that some user during the experiment found a possible use for architecture representation picture production we focused on this use in order to experiment on a restricted case how much implicit information was handled by the system. The first parameter

Indeed, with the artistic style, the expectation of the user on "how the artistic style should be applied" is hard to define a priori. Before running the neural style code, what produce this architecture building in a Van Gogh Style is hard to figure and takes time to materialize. On the other hand, on more precise task that still require some licence in decision taking, the expected output is easier to materialise with actual digital design

FIGURE 5.5: Regarding the input picture A and B we get with default option on neural style this output

tools. For instance, from the content image C, the user will try to produce with fewer inputs a picture similar to the picture 5.7. It is important to take in consideration that theoretically, the user, in a seek for CR will not aim to have exactly the picture 5.7. If he can precisely foresee exactly what he wants, he will not be interested by such approach that aim for CR in their design process.

FIGURE 5.6: Base Image for the Neural Style tool

5.2 Scale of change, indirect valuation of the implicitness

This use of Neural style offers us a rare case where measurement is possible through the comparison of input and output. This measurement aims to have an indirect index of the CR of the tool. Nonetheless, since the implicit is by nature something hidden, it is important to clarify what are the implicit information of change that are pre encoded in the tool. Indeed, if we compare with a filter effect tool in GIMP it is possible to state that implicit information made desired change without knowing how much change are desired. Indeed if only a small part is change base on implicit information it is not a

FIGURE 5.7: Sample taken in the sea of expected outcomes

Tentative	base line	ignore color	ignore anti-aliasing	number of texture
Substrate to 1	98.07%	74.22%	13.53%	1
Substrate to 2	96.03%	60.73%	5.95%	3
Substrate to 3	95.83%	55.48%	6.67%	4
Substrate to 4	80.91%	54.98%	10.12%	4+
X	76.31%	48.99%	9.31%	6
P	75.51%	46.48%	9.26%	6
Substrate to goal	42.88%	35.08%	12.34%	6

FIGURE 5.8: According to the input style, we can see the difference from content to output

valuable ratio for a tool that is supposed through its intelligent algorithms to help the designer in intellectual tasks. In order to compare the goal picture and AI generated outputs we use an image comparison tool (Figure 5.9) that compare the values of each pixel of two pictures. By this means, the tool display a percentage of difference and highlight the pixels that are different between the two pictures.
The percentage of pixels that match the initial expectation is called the "Expectation Delta".

5.2.0.1 Iteration parameters setting

Since the process is slightly similar to Deepdream, there is intermediary images that are use to be resubmitted in the ConvNet. This parameters is usually set on 1000. Since the goal is to get creative insight, a perfect rendering of the image is not aimed. Nonetheless, the production value of the tool has to reach some production value threshold in order to be readable by the user. As we can see in 5.10 the readability evolve gradually and it is hard to clearly set a threshold of it. To decide a number of iteration that does not takes too much computation time in regards to the flow. We measure the distance in similitude from the output to the sample of goal in 5.11 . To put this distance in balance

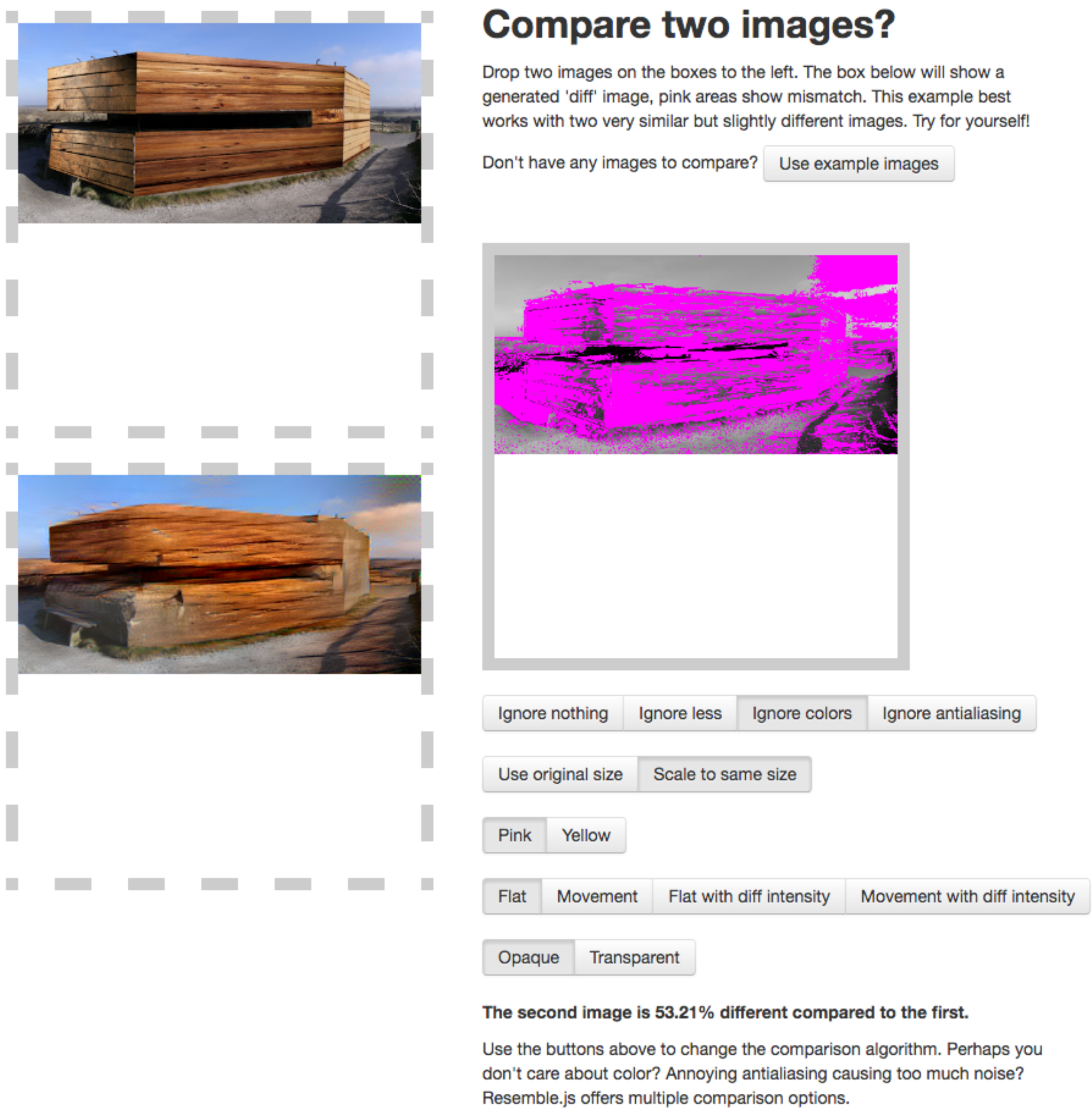


FIGURE 5.9: This system has been used to compare the pictures

we indicate the value of the remoteness to the original image. It attests that the designer was aiming to a substantive change.

We can see that their is various resemblance depending of the parameters taken in consideration. Since the jpeg encoding of the picture is destructive, small changes in the pictures modify lots of pixels value in order to reduce the file size. The base line can not be taken as reference because it shows to small delta. Ignoring anti-aliasing seems to be a good parameters since the algorithm avoiding anti-aliasing is partly responsible of lots of pixel value modifications since the images are unstructured patch of texture. As we see in the 5.12 .

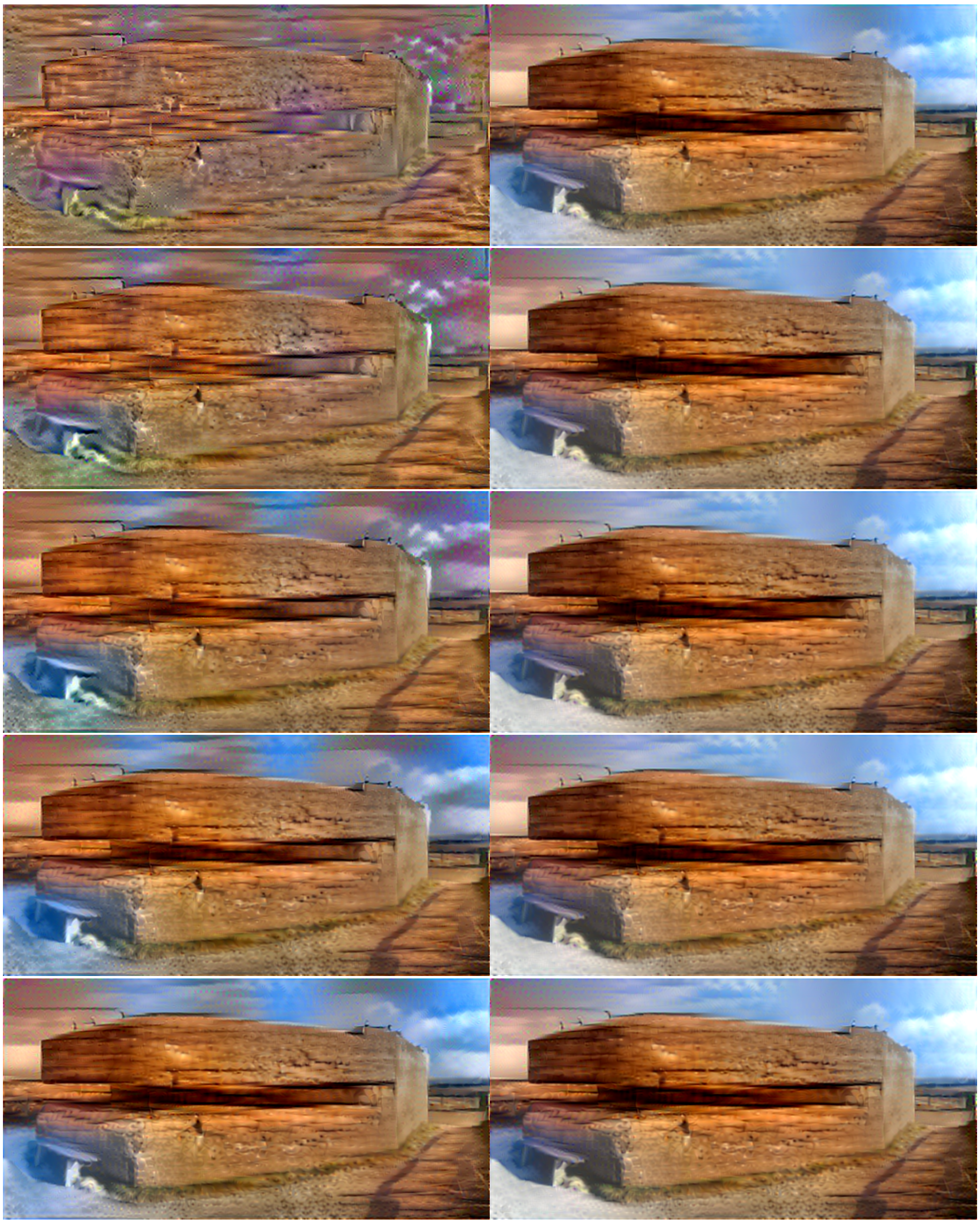


FIGURE 5.10: 10 iterations on the same pictures of the texture set n3

5.3 Indirect measurement of CR

As seen in the previous part, a restricted use of Neural style as tool to apply texture can be transformed as a experiment to test the implicitness. As seen in the Figure 5.12 between 500 and 1000 iterations the output matching percentage with the expected goal do not increase much so keeping 1000 iterations as a fixed parameters for all the following experiment ensure a fair comparison of measurement of CR. Since the goal is set, we will vary only the value of the input style and see how much it impact on the proximity to the goal.

Difference	base line		ignore color		ignore anti-aliasing	
Distance from	origin	goal	origin	goal	origin	goal
Iteration 1	97.87%	98.64%	81.11%	83.30%	30.37%	31.52%
Iteration 2	99.30%	98.20%	78.53%	83.50%	26.01%	28.78%
Iteration 3	99.30%	98.04%	75.17%	81.63%	18.42%	23.01%
Iteration 4	98.57%	97.04%	69.79%	77.12%	11.18%	17.18%
Iteration 5	97.61%	96.00%	63.39%	71.46%	8.11%	14.70%
Iteration 6	96.32%	94.79%	57.91%	66.69%	7.32%	14.47%
Iteration 7	96.14%	94.75%	55.87%	65.33%	7.14%	14.70%
Iteration 8	95.89%	94.54%	56.03%	65.71%	7.14%	14.89%
Iteration 9	95.94%	94.70%	56.30%	66.21%	7.00%	14.91%
output	96.04%	94.83%	56.34%	66.39%	6.87%	14.88%

FIGURE 5.11: distance from the content picture and distance to the goal picture (input style 3)

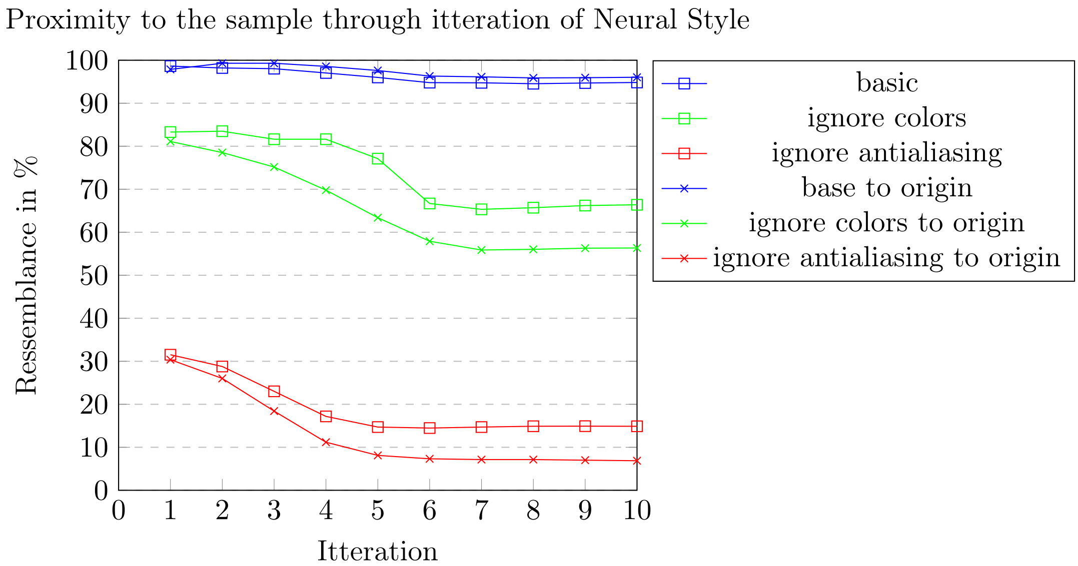


FIGURE 5.12: The most high rate of modification of the resemblance rates occurs at the 500_{th} iterations

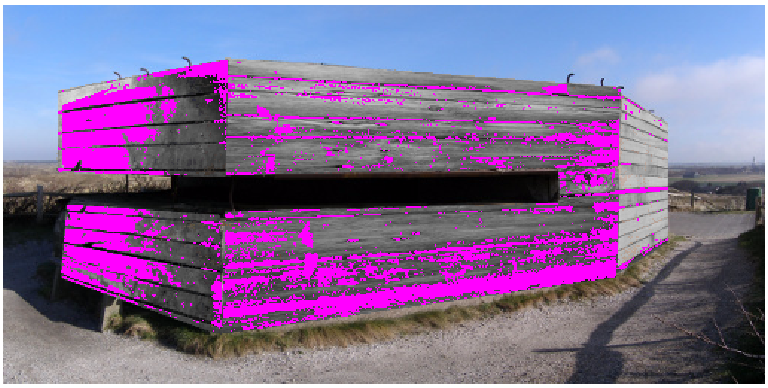


FIGURE 5.13: The designer is aiming to a change of 12.34% from the original picture. Those change are located on the texture building

5.3.1 Define a goal of an open goal process

The use of Neural Style for texturing is somewhat paradoxical since the Designer is not searching for a pure ideation output but he or she doesn't have a clear idea of all the detail of the image. He or she wants something that look realist without having to create it from end to end. In order to measure if what he or she sees look what he or she expect we can't rely on a simple interview because of bias. They could as much answer answer that the picture look like what they expect because they find it inspiring or nice as much that they can reject it because they didn't approach the use of the system with an open design attitude. The attitude of their approach of the tool is, as we see in chapter 3, subject to even more bias question. Thus we ask them to define a goal even if it was not exactly their goal. The image of an average or approximative goal can serve as reference for comparison of the different outputs created during the interactions with Neural Style. As we can see in Figure 5.16 and Figure 5.15 that is the output of comparison software. The pink pixel allow to visualize between two interactions, the amount of discrepancy with the goal image.

Distance to goal through iterations. (case 3)

5.4 Implicitness in tool input message context

The following figures (Figure 5.17) show the different attempt to use Neural Style as a fast texture mapping system. We took this case because the user, unlike the case of the "nest bunker" (Figure 5.14), was able to produce a sample of his kind of expected output. The tabular Figure 5.18 show that above threshold of input information the closeness to the goal doesn't improve much. To compare we use Resemble.js[1]

5.4.1 Experiment protocol

In order to verify that this use of Neural Style as a fast texture mapping was not limited on some type of content or some type of texture we conduct an experiment where it was asked to Designer to collaborate with Neural style in a similar way that Designer H did. Through an ideation process [77] typical to project open ended starting attempt, experiment interaction with AI. The experiment is divided in 4 stage.

- Warm-Up: Visual Turing Test

[1] `https://huddle.github.io/Resemble.js/` Created by James Cryer and the Huddle development team

(A) (A)

(B) 6 textures (B)

(C) 6 textures (C)

FIGURE 5.14: Another attempt to use Neural style as a smart texture mapper

FIGURE 5.15: With very few information provided the designer H reach difference of 11.03% with one sample of his sea of expected outcomes

FIGURE 5.16: With very few information provided the designer H reach difference of 9.47% with one sample of his sea of expected outcomes

- Goal Picture photo-montage
- Interaction with Neural style
- Designer's and AI output evaluation

In order to avoid the problem similar to the one encounter wit the case showed in Figure 5.14, we asked to the designer to produce a picture of their goal before interacting with Neural Style. Since previous testers had lots of experimentation, there was a risk of bias in their way to select the content and texture to apply on it.
11 Designers have been contacted to conduct this experiment. They are of different gender, country and level of education. We have tried to diversify as much as possible the selection of designers for the experiment. Indeed, a panel of testers exclusively made of male may be the cause of the phenomenon we will observe trough this experiment. Since those designer were not as skilled in programming and didn't have computer knowledge necessary to make them use the tool by themselves on their computer, we proceed for them the file they wanted to send to Neural Style system. For similar reason to their possible lack of knowledge about Computational creativity, they had to do a warm-up quiz as introduction to the experiment.
As said previously, the goal picture is required because it is the only mean to have a reference for comparison of the picture with Neural Style. Nonetheless, this is not a perfect representation of the designer's intentions. Indeed, his or her limited skills in drawing or knowledge of photo-montage software may restrain him or her ability to represent his or her design intention. Thus the goal pictures have to be considered with prudence: this is not a mirror image of designer's intention mental image. They even may have no clear mental image at all. Since the brief of the exercise is described as a draft attempt in a more general design process, it is not expected that the designer had a precise image of what he wants to achieve. In fact, in the brief of the experiment a

(A) One texture of wood plate

(B)

(C) 3 textures

(C)

(D) 4 textures

(D)

(E) 2 textures on the background of the content picture

(F)

(G) 6 textures

(H)

FIGURE 5.17: Outputs and their corresponding input style

Tentative	base line	ignore color	ignore anti-aliasing	number of texture
1	95.59%	71.61%	20.07%	1
2	94.39%	68.94%	18.81%	3
3	94.83%	66.39%	14.88%	4
4	74.93%	53.39%	9.47%	4+
5	69.52%	48.20%	10.00%	6
P	69.08%	47.30%	9.82%	6

FIGURE 5.18: The different output created with different input style show that up to 4 texture in the input the output similarity doesn't improve

scenario is proposed and it states that the designer apply this texture on a building to create is own visualisation to know *"how it looks if..."*. This approach is thus oriented through a step of a creative research of shape and material rather than fine rendering of an image for communication with an imaginary client. This precision is important since it induces the designer in an *open design attitude*. If he tries to achieve that as a production exercise he or she will spend too long time to create the goal picture. Above that, regarding the non-specialisation of Neural Style for architectural texture layout, it will be unfair to compare production with image made at a rendering level. In a real design practice, a designer will not spend too much time to get a perfect result for an image production just intended to "feel" how looks like visually a vague idea. Rendering image are for final presentation of the project while sketch are useful to open path of design exploration through constructive perception [35] or as rapid means of communication with other designer in the collaboration.

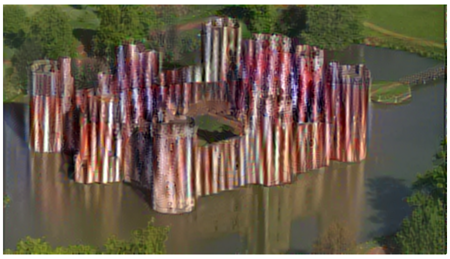
Once this photo-montage is created, designers are asked to create the first input message image (IMI) for the Neural style system. What is called the IMI correspond to the style picture in normal use of Neural style. Since the way to create this input message is not intuitive, a brief have been provided to the designer who participate to the test.

On a jpeg file of 500 by 500 pixels or of a size equal to the size of the original image put samples of texture needed to transform origin picture into an image similar to the goal picture. The AI system will automatically modify and map those textures on the relevant position and shape above the origin picture. Therefore textures on the input message image don't need to be placed on a specific position or cut in a specific shape. Nonetheless, it is important to remind that all the texture of the original image will be replaced by one of the input message image textures. Therefore, it is necessary to provide existing textures of the background that are not intended to be changed. As you can see in the three samples of input message image, a sky texture is always provided. The input message image is constructed that way in order to keep the sky blue in the final output. Since the goal is to apply a material on the building, not to make an artistic rendering of the original picture it is advised to proceed in this way. The purpose of this input message image is to provide enough information to the AI in order that it proceed a design task similar to *the one conducted in the goal image construction part. This input message image will be processed by the AI and the output image file generated will be shown to you. Based on this feed back and in order to get an output image closer to his expectation, you adjust or modify the input message image. This interaction with the AI will be iterated three more times.*

After the four interaction a questionnaire is submit to the designer. According an *Inspiration* scale and a *Expectation* scale, the designer is asked to rank the 4 outputs generated by neural style. After the two rankings established, the designer is asked to insert in this ranking the goal picture. The aim of this question is to see if there is a common ground that can be established between AI and designer during design collaboration. Indeed, this evaluation shows if collaborating with AI can bring improvement in the design process through outsourcing representation task. Thus, if a goal picture is not ranked first, it can be a sign of CR through the recognition of intelligence as understanding the implicit context of the original picture and the input message to create a readable enough pictures.

Expectation is the evaluation through ranking. Output image matching original in-

(A) first output ranked first



(B) second output ranked second

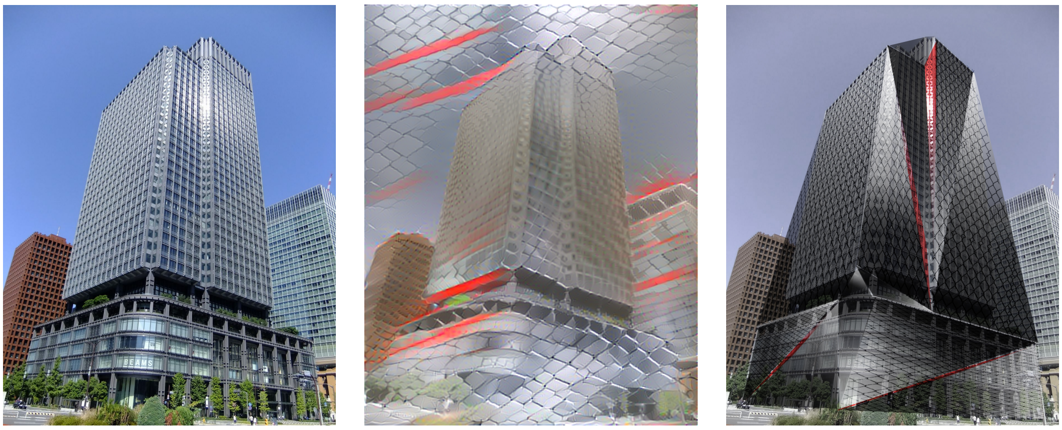
(C) Goal image ranked third

(D) third output ranked fourth

(E) fourth output ranked fifth

FIGURE 5.19: The 5 images created by Designer and AI ranked on the expectation scale.

tention regarding the initial idea is ranked first. Then the other outputs are ordered according their success to fulfil expectation compare to the other. It is important that the reference used to compare image is the idea they had before starting to make the "goal picture". It is important to dissociate the original intention from the goal picture because, as previously mentioned, designers' lack of skill in image manipulation software may have restrained their ability to produce the image they wanted initially. Above that, creating the goal picture may have change their original intention. As we can see in the Figure 5.19c, the time constrains and the challenge level implied by the couple [original image / textures to apply on] may lead to create a goal picture that does not match with the designer's original intention. Indeed, in Figure 5.19 the goal pictures (Figure 5.19c) arrive only third in the expectation ranking. It shows that Designer A wasn't clearly aware of what he or she wanted before visualising it, or discovered relevant features he or she didn't have time to materialise in the photo-montage. For this reason, the goal pictures need to be approached as just an approximative instance of designers original plausible intentions. This *Expectation* evaluation is similar to the subsection 3.1.2 seen in chapter 3. *Inspiration* is the valuation designer place upon the output as visualisation



(A) The designer C select this image as a starting point of the design task

(B) Those fluid layout of the wire-mesh on the building shape was considered as a potential lead to continue the design task by designer C.

(C) Initially, designer D was more envisioning such way to layout wire-mesh on the high-rise.

FIGURE 5.20: The designer C said being inspired by this output generated during the second interaction with the AI.

document that may help them to step forward in their design research. This is the "creative potential" they found in the picture (Figure 5.20). It can be an unexpected glitch, shape, colour, layout that deviates their design process from original intention. The recognition of inspiring qualities in the output pictures or goal picture is related to the notion of constructive perception [35]. The designer is able to reorganise his or her own perception in a different frame of mind than the original one he or she used to approach the design task. The Figure 5.19 shows a case where designer find out qualities despite

Tentative	base line	ignore color	ignore anti-aliasing	number of texture
1	97.62%	82.92%	33.55%	1
2	93.39%	76.62%	20.57%	3
3	92.84%	71.81%	16.79%	4
4	84.33%	60.73%	12.02%	4+

Figure 5.21: Similar to the previous experiment, the ConvNet is replaced by a ConvNet trained on 2 millions pictures of place compare to 14 millions with the VGG

the image is not matching is the goal picture. Indeed, As we can see in Figure 5.19c, the designer roughly delineates the window area but didn't think, or had time, to create the reflection of the new material in the water. The AI meets Designer A implicit intention of realism in Figure 5.19a.

5.5 Training as source of implicitness

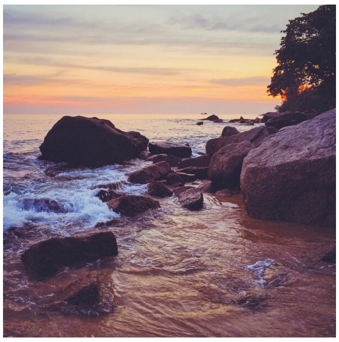
The images and data harvested through this experiment allow to make overlap. The result of the visual Turing test[78] can inform on the intuitive understanding of designer. Referring to the criteria of "Evaluation methodology of digital tool for design" in chapter 2, we could relate the designer's score of the visual Turing Test to its "Conceptualisation - Understanding" couple of the tool. Since those criteria measurement are in the designer's mind, they are hardly accessible. According to the Figure 2.7,those criteria are related to "Expressivity" as causes and "Implicitness" as result thus by measuring exterior indices found in causes and effects, we may discover correlation.
On a similar approach of indirect measure that can give access to a measurement or evaluation of what is on designer's mind while interaction, the criterion of "Implicitness" have been explored through a concept of "Expectation Delta". As show on Figure 5.23 base on a variation of Figure 1.13, the Expectation Delta (ED) is the difference between what the tool user expect as effect of the input message will get on the output production. On case of neural style, users may have a vague idea of what should be an artistic rendering of a content image in the artistic provided style. By looking on website such as `http://ostagram.ru`, `https://neural.0101010101.com` or `https://deepart.io` we can found user owning the tool in order to produce visual that take profit of the system's principle with input style images that differ from the impressionist set of the original paper example. In fact, the sum of all those examples gathered on the entry point of this web-service offer to each user the cumulative experience of all the previous ones. This dynamic could be interesting to study but unfortunately two critical parameters are missing: user's intention and their satisfaction regarding the output. Those elements are correlated according the diagram presented in Figure 2.7 and just getting the input

and output file even in large quantities, don't provide access to necessary informations for deeper analysis.

5.5.0.1 Obstacles to the analysis

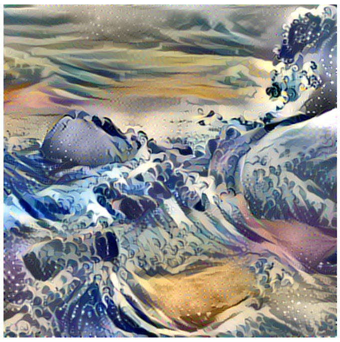
As we can see on Figure 5.22 taken from a website sharing DeepArt (Art made with ConvNet), the common use of Neural Style program to generate images is made with an aesthetic frame of mind. The goal is to create images for a recreational purpose and the sole sake of aesthetic curiosity. Users may have no conscious goal when they select the two pictures to integrate and are more curious to see what "happen if...".

In such "artistic" use of Neural Style, the output cannot be compared with a reference image. Indeed, objective measurement is not possible with a purely artistic or impressionist goal because there is no clear goal picture that can be materialised upstream before generating the image. Above that, when their is not a clear frame of perception reading the depiction of the image correspond to the meaning [79] given by the picture interpretation. Since the ability of the user rarely match with the painter's ability whom they borrow the style. It is hard to ask them to draw their vision if they are not able to achieve a certain level of image quality that ensures an objective comparison with the original model. Photo-montage is a much easier creative task that can be achieve by a lot more people than drawing. The design task of changing the material is indeed easier and done in "bricolage" state of mind[80]. The "bricolage" design process relies on the constructive perception in order to overtake the appearances of an unpolished rendering. For the reason above mentioned, with an ordinary use of Neural Style, the evaluation of output quality through meeting expectation or inspirational value is difficult since only the designer knows what he or she wanted to create and only him or her can judge the output. This judgement is depending on whether designer mood arbitrary change. In most of the case ©Photoshop or GIMP have been used. Those software are photo-editing tool-kit that proposes tools to clone pixels values of an area to another one, skew, distort or layering images above each other. Therefore, it offers various strategies to complete the task. To impeach testers going too much into the details, the task duration is limited to 20 minutes. Above that, impressionist pictures do not belong to a shared reality as much as a photographic picture can achieve. There will always be a freedom of interpretation that makes evaluation difficult to conduct.
For this reason, the fast photo-montage was the appropriate type of exercise since it presents a balance between explicit representation, loosely skill demanding and not prone to create demanding expectation on image quality to achieve comprehension.



(A) Image selected by an anonymous user. In the experiment, the image used as content image is the original image used as bases of the photomontage

(B) The famous wave of Hiroshige is here used as a style sample for neural style



(C) sample of output produced by anonymous user of online version of neural style

FIGURE 5.22: Sample of the use of an online version of Neural Style find on http://neural.0101010101.com

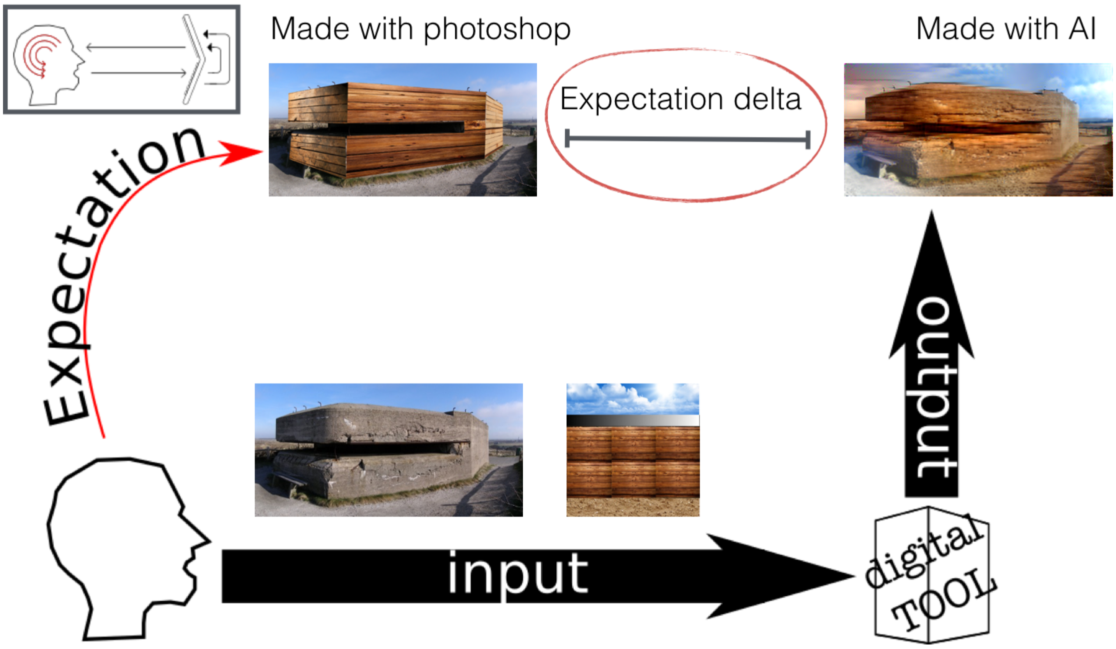


FIGURE 5.23: The difference between what the designer's design intention and the output generated by AI partner

5.5.0.2 Expectation Delta through attempts and Input Message Image (IMI) specification

As the Figure 5.8, the specifications in the IMI can vary and the ED score may vary according to it. With those multiple case, we can explore the IMI construction and its influence on ED. The measurement of the specification is harder regarding the variety of strategy to create IMI. Indeed, it is easy to count number of texture but in some

Designer	IMI 1	IMI 2	IMI 3	IMI 4
Designer A	2	3	4	5
Designer B	2	3	4	3
Designer C	4	3	5	6
Designer D	8	8	7	7
Designer E	1	2	2	4
Designer F	5	5	6	4
Designer G	4	6	7	6
Designer H	5	6	7	3
Designer I	4	6	11	12
Designer J	4	7	13	17
Designer K	1	3	5	4

FIGURE 5.24: Each Designer and the IMI specification of the four interaction of the experiment step

case, Designer get the idea to change contrast of some texture or to use as a background the original image and all the texture "naturally" embedded in it. Thus we decide to count textures without regarding their size or layout. If a pattern or texture is created in the layout of texture, such "meta-pattern" have been counted as a specification. When original image is used as background image, each area that contain significantly different texture has been counted as a texture specification. Ideally, we would like to measure the exact amount of information in kilo bits that is involved to create the IMI. In such a fashion that it takes "amount of information, noise, channel capacity, and rate of information transmission" [81] we could imagine to very precisely measure the input and output between designer and computer. The Figure 5.24 show the measure of each IMI despite some IMIs were difficult case to measure a quantity of specification intentionally provided by the designer. Indeed, the relativity of their understanding of Neural Style internal mechanics prevent them to always make significant specification for the machine. For this reason, we decide to count the specification from the designer point of view. For this reason, some IMI specification measure are very high. Designer had sometimes tried very convoluted process to create the IMI.

The connection between the IMI Specification and the ED score in the other case doesn't confirm the correlation seen in the section 5.2. Indeed, some case of original image used as content shows that whit surface are not very well detected by the ConvNet. Texture of the IMI don't appear on the expected surface when this surface contain lots of white colour. Thus, the increase of specification don't change proportionally the ED score.

5.5.0.3 Expectation Delta ranking and Designer's ranking

Another possible overlap of experiment data is to look if the subjective classification of designer match with the ranking of expectation delta. By this way we can observe if a measurement method can be substituted to designers' interview. As the Figure 5.25 shows, only 4 cases on 11 are matching. Despite that regarding the probability to have those matching 4 times are a little above than a chance average, we can't conclude that the ED score relate the expectation of designer. It is important to note that in some case, the ED ranking is inherently flawed. For instance, the case of the Designer A show that the ED rank the attempt 1 in last while the designer rank it first. Such large gap of evaluation is easily explainable by the fact that the goal picture, used as reference for ED score is rated only third by the designer. Indeed, he explains us that the output 1 shows reflection of the texture in the water. A point he didn't think or had time to represent. It is part of his expectation that the AI complete the image for visual coherence. Thus, the use of goal picture as reference for ED calculation can be discussed. Originally, he didn't think at all about the reflection and that the output who remind him that. Thus, can we say that coherence of the picture was really is original intention and thus expected even at the first interaction ?
Those kind of question are typical of studies that involves questionnaires. It can't be overcome in a indubitably fashion. Nonetheless, some redundancies can give clues on the next step for interaction experimentation. For instance, we can notice that the first attempt is ranked last 15 times in both ranking. Since it's the first attempt, except the case where the Designer obtain the best of the system at the first attempt (Designer A, D, F, G) we can understand that as a sign of progress for most of the Designer in their interaction. As corollary of this observation, we can state that the method is good enough to detect the progression worst output. But as we will see, high ED can't be compared because of the vagueness of the measurement tool with initial settings. Indeed, if we look at Figure 5.26 we can see that some ED scores goes beyond 40%. In the initial case, measurement setting have been chosen because they allow more variation and thus are easier to compare for value that fluctuate between 4% and 35% (Figure 5.8). Nonetheless, even with those relaxed setting, some score goes beyond 62% despite the pictures present an air of family. Pictures that are clearly different can have a difference score of 45% with those loose setting. Thus, above 30% the score can't be use to compare the ED between pictures. Other means or setting of the tool need to be used.

5.5.0.4 Best Expectation Delta with different ConvNets

Neural Style is based on a pre-trained ConvNet that provide a network of tuned function that is used to define the layout of the different texture elements of the input message

Designer	Expectation	ED
Designer A	12G34	4321
Designer B	G4321	3124
Designer C	G3241	3241
Designer D	1G342	3142
Designer E	4G321	4321
Designer F	1G324	1234
Designer G	G4321	1234
Designer H	3G214	3214
Designer I	4G321	2341
Designer J	G4321	3421
Designer K	G4321	4321

FIGURE 5.25: The comparison of Expectation ranking and the Expectation Delta Ranking

into the original image. In fact the system doesn't apply modification on the original picture but, base on the input images provided, define a sort of neurons set activation targets and modify pixels of a base image. Then, it calculates how much those modifications are activating the target neurons.

The network of tuned functions is built through the training of the ConvNet and therefore, they fit to a specific task. Usually those tasks are recognition tasks. A network can be trained to recognise different dog race or different car models. According to its recognition speciality, artificial neurons are tuned differently in the ConvNet. The ConvNet can be seen as a database of tuned neurons to detect specific features that indicate the belonging to a category. Thus, since in Neural Style this "ConvNet database" determine the procedure to apply the artistic style on the picture. Changing the ConvNet at the core of Neural Style systems produce with the same input parameters, a slightly different output than the one produced with the default ConvNet embed in the system. The default ConvNet is VGG19. This network is designed and trained by the Visual Geometry Group of Oxford University [82]. This ConvNet contains 19 layers and have been trained on the Image net data-set for the ILSVRC -2014 challenge (ImageNet Large Scale Visual Recognition Competition). As comparison another ConvNet such as Alexnet-Places205 have been trained on the MIT Places data [83]set and present an AlexNet architecture [84] trained with the Places 205 version of the data-set. 205 refers to the numbers of label categories.

On the figure Figure 5.27 we can see that the Expectation Delta of some experiments case created with different ConvNets at the core of neural style. We select three ConvNets with different architecture or trained with different data set.

Designer	ConvNet	IMI 1	IMI 2	IMI 3	IMI 4
designer A	ImageNetVGG19	10.89	9.5	6.5	4.97
	ImageNetVGG16	16.21	13.76	11.09	9.85
	PlacesVGG16	17.66	16.19	14.44	10.68
Designer B	ImageNetVGG19	13.55	31.37	11.62	44.1
	ImageNetVGG16	29.89	44.1	23.65	48.5
	PlacesVGG16	28.32	49.12	28.46	54.94
Designer C	ImageNetVGG19	78.42	34.06	33.53	41.34
	ImageNetVGG16	85.01	42.94	46.27	42.55
	PlacesVGG16	90.79	42.88	41.5	56.83
Designer D	ImageNetVGG19	18.5	27.43	17.87	19.89
	ImageNetVGG16	16.19	29.32	16.17	16.21
	PlacesVGG16	21.24	38.51	16.66	18.2
Designer E	ImageNetVGG19	41.85	28.98	14.81	14.53
	ImageNetVGG16	43.97	35.55	25.33	35.34
	PlacesVGG16	43.07	44.46	48.37	39.5
Designer F	ImageNetVGG19	8.13	8.52	11.88	69
	ImageNetVGG16	61.96	50.18	45.96	70.56
	PlacesVGG16	54.68	49.32	33.62	72.09
Designer G	ImageNetVGG19	14.98	16.23	18.85	19.17
	ImageNetVGG16	31.36	27.31	25.39	52.83
	PlacesVGG16	25.22	21.96	25.08	39.98
Designer H	ImageNetVGG19	22.69	20.56	10.93	25.75
	ImageNetVGG16	41.28	34.67	36.01	49.9
	PlacesVGG16	35.15	38.44	23.8	31.65
Designer I	ImageNetVGG19	31.75	20.95	26.23	29.45
	ImageNetVGG16	58.35	45.88	43.03	46.79
	PlacesVGG16	48.43	41.43	35.35	46
Designer J	ImageNetVGG19	52.53	47.53	40.36	42.71
	ImageNetVGG16	56.31	58.39	54.03	53.96
	PlacesVGG16	62.72	56.76	51.1	65.48
Designer K	ImageNetVGG19	20.7	18.81	14.88	9.47
	ImageNetVGG16	23.04	21.76	22.03	14.13
	PlacesVGG16	33.55	20.57	16.79	12.02

FIGURE 5.26: Expectation Delta for each Designer calculated on images produced with 3 different ConvNet at the core of Neural Style program.

5.6 Cases

In this section the 11 designers data are presented.

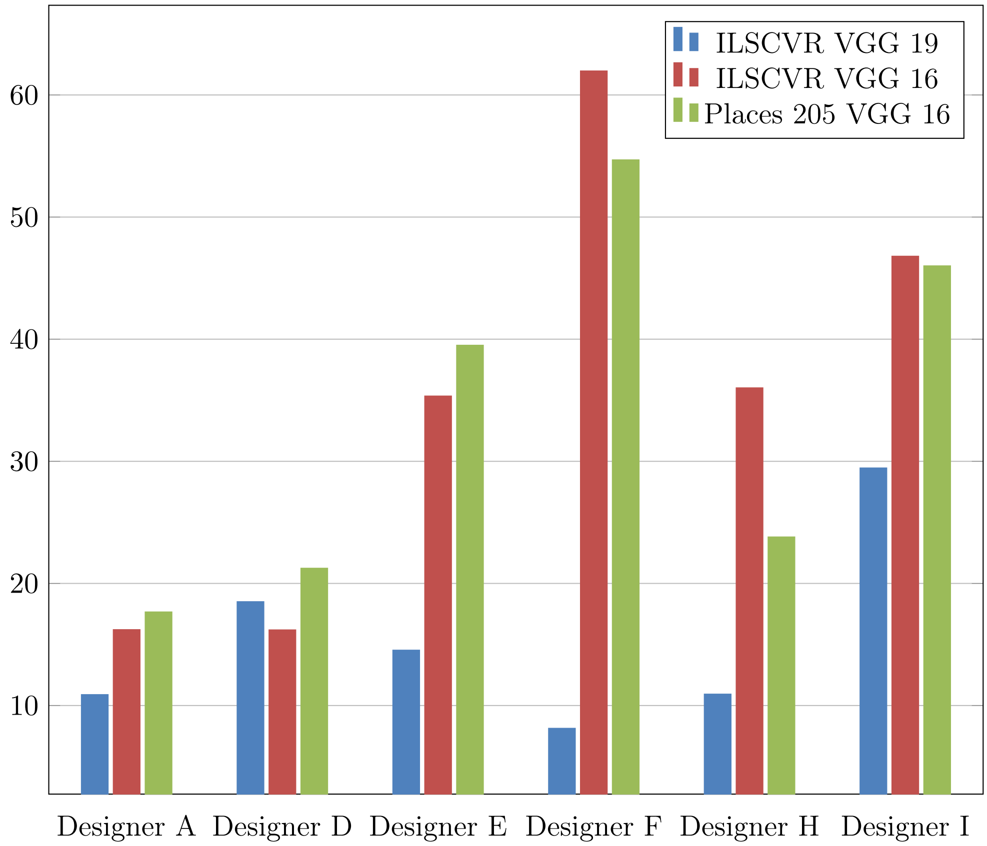


FIGURE 5.27: ConvNet impacts on the expectation Delta

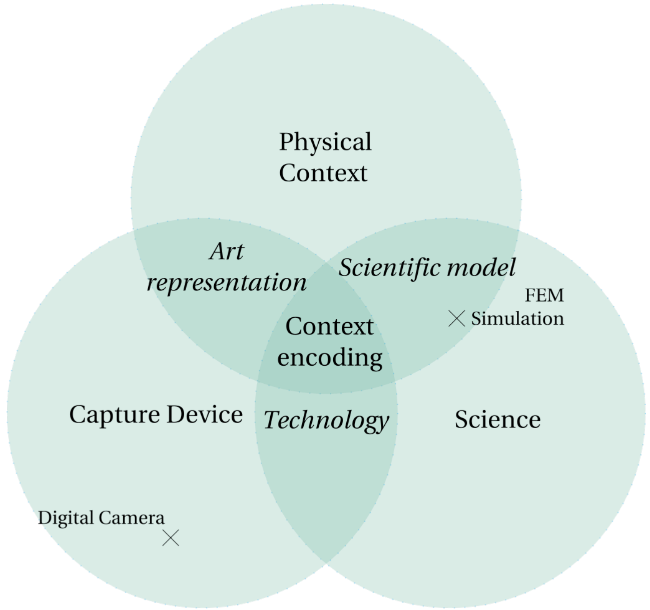


FIGURE 5.28: The encoding of the context have been done until know, whether using intellectual resource to extract the "intelligence" of the context whether automatically extracting pure data without intelligence through sensors.

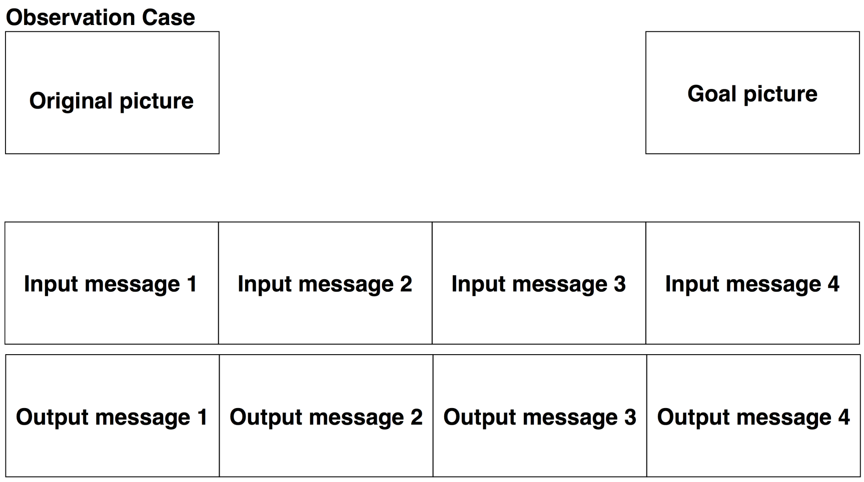


FIGURE 5.29: General Layout and Legend

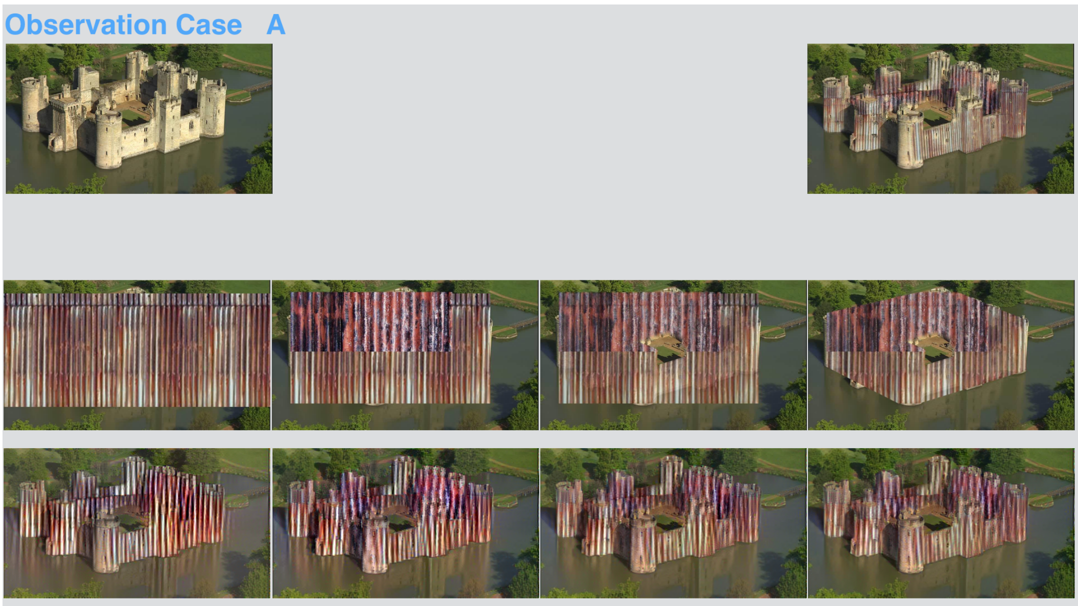


FIGURE 5.30: Designer A

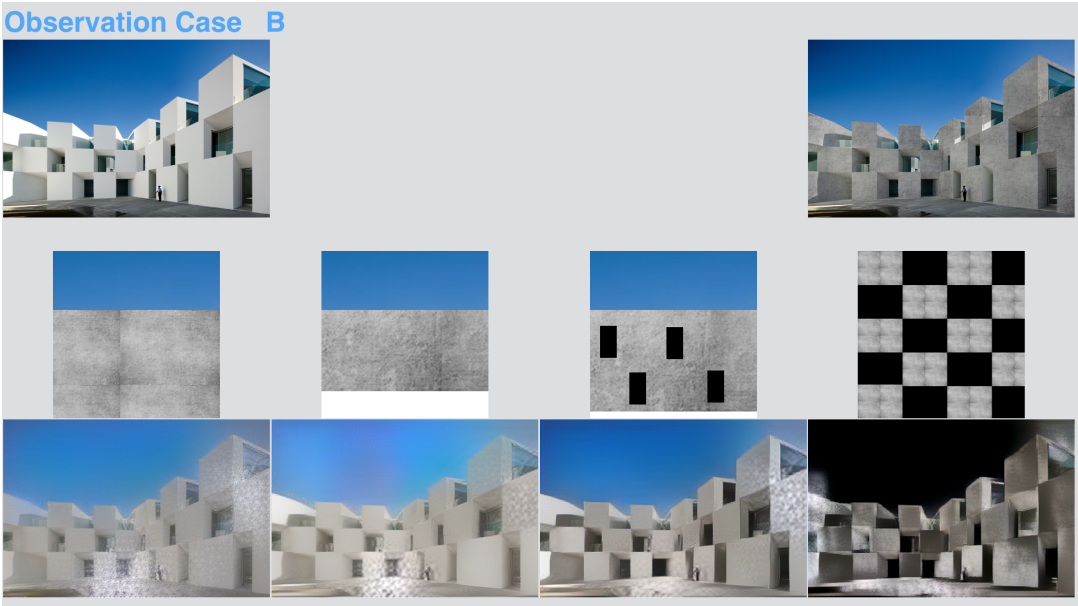


FIGURE 5.31: Designer B

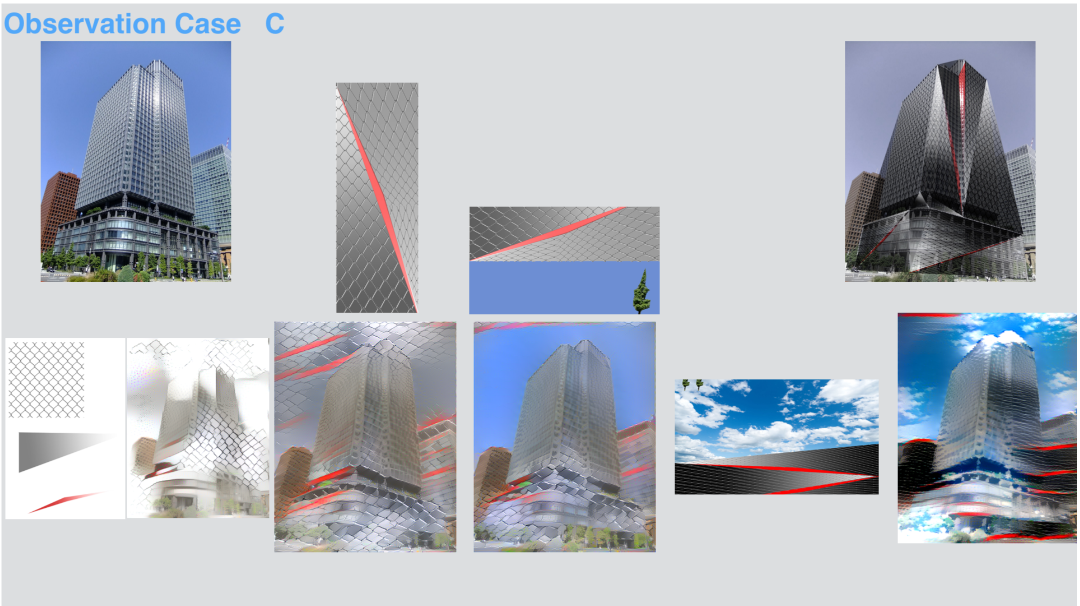


FIGURE 5.32: Designer C

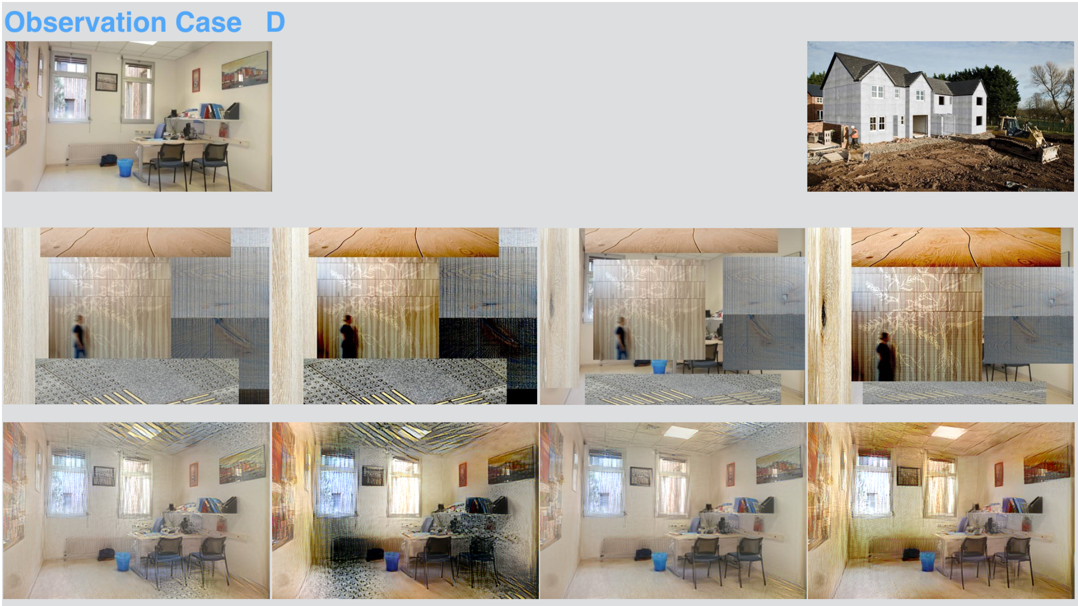


FIGURE 5.33: Designer D

FIGURE 5.34: Designer E

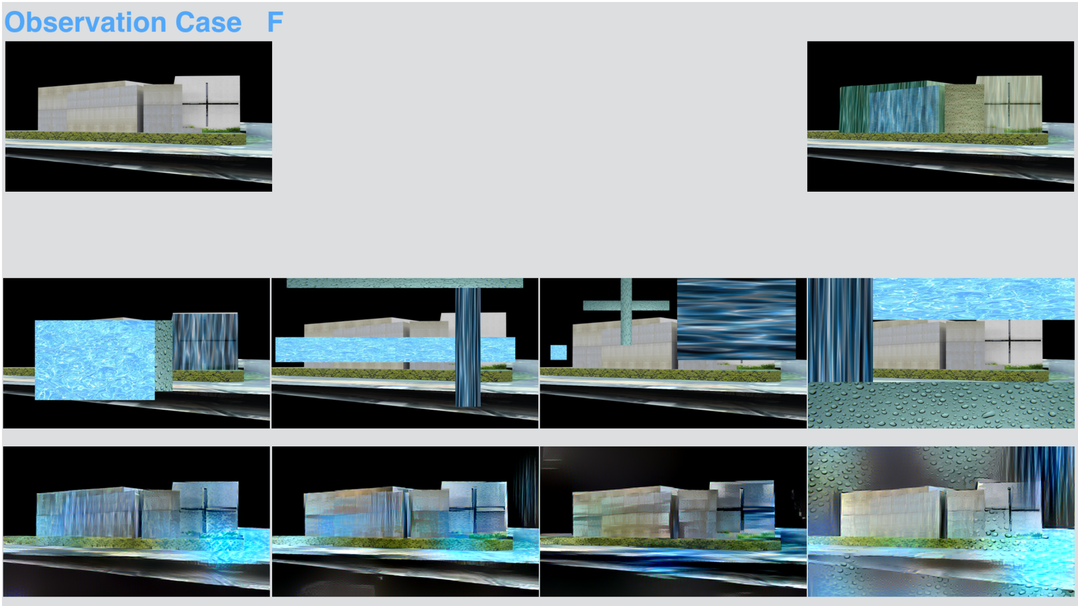


FIGURE 5.35: Designer F

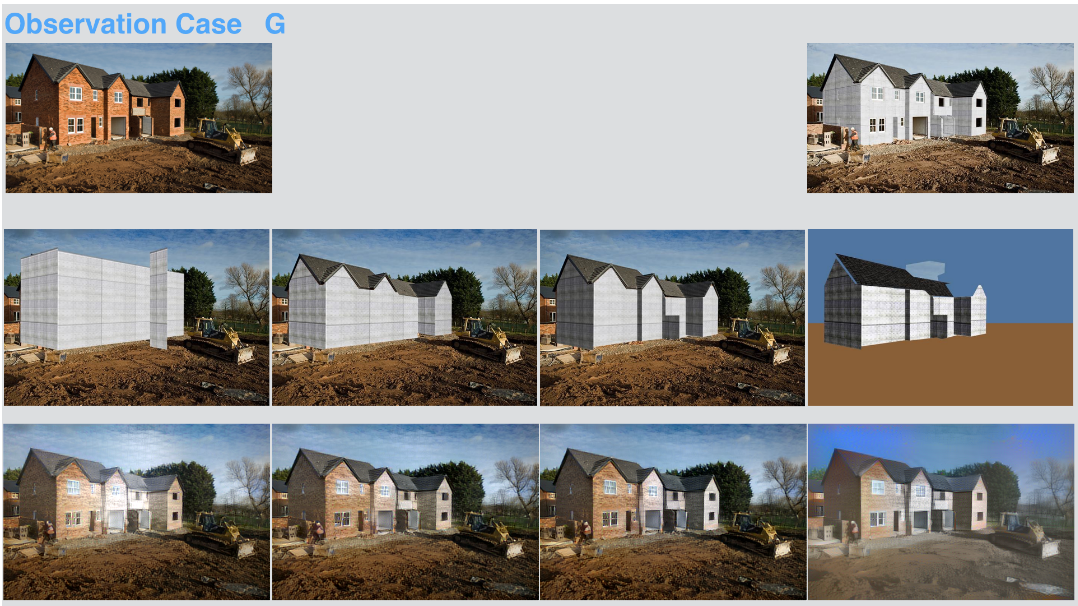


FIGURE 5.36: Designer G

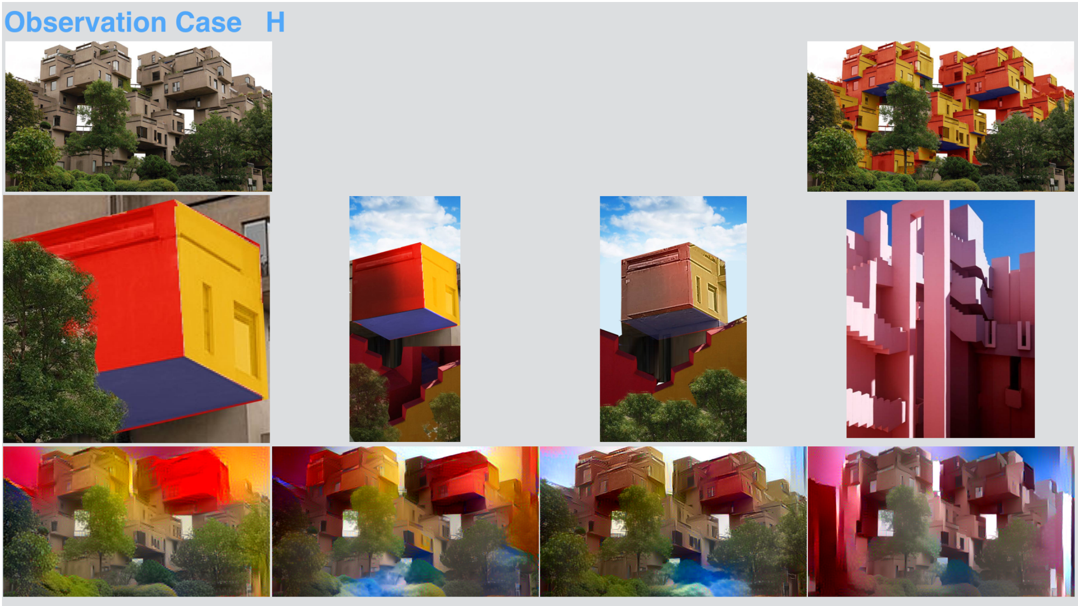


FIGURE 5.37: Designer H

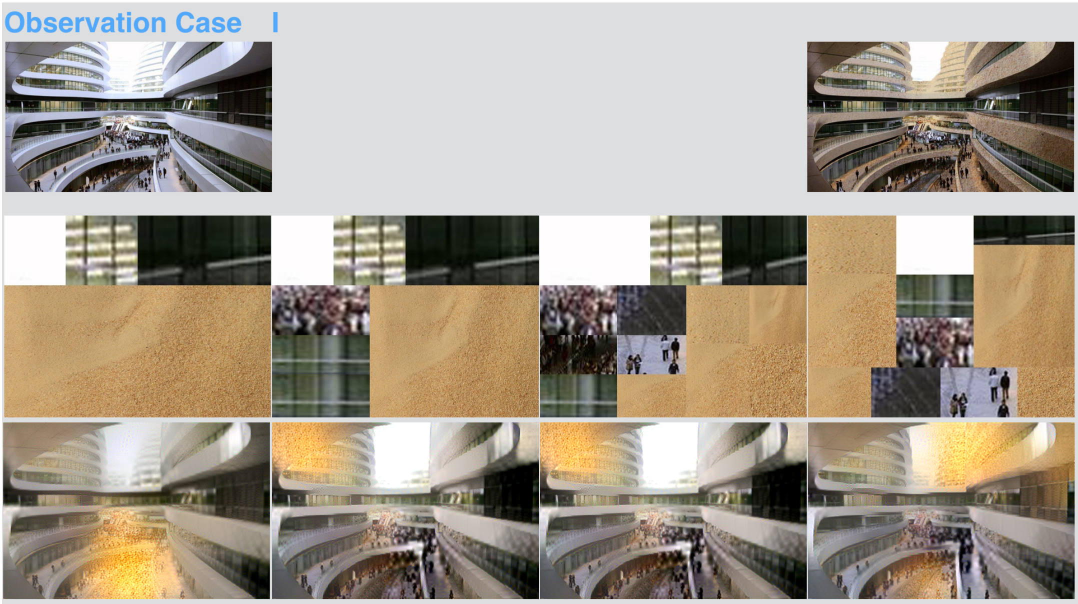


FIGURE 5.38: Designer I

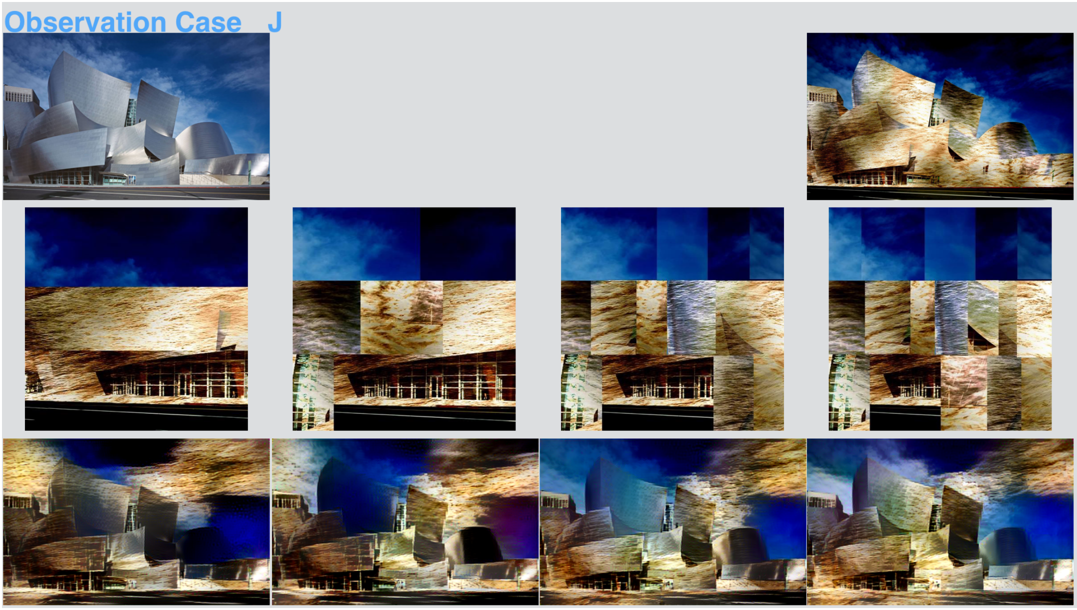


FIGURE 5.39: Designer J

FIGURE 5.40: Designer K

Chapter 6

Conclusion

Figure 6.1: Specific attitude, suitable flow and encoding of the context are necessary in order to reach a resonance frequencies of Compute-Designer oscillation in case where the designer is not an expert in computation

6.1 Summary of the Result provided through the different experiment

Through three experiment this research measured various parameters that seems to be related to CR regarding the Evaluation of digital tool made previously. This tool evaluation system put us on the lead to build experiments that can allow us to observe features and parameters of the CR. Those parameters and features have been measured or when not possible dissected into observable features that combined together may relate the presence of this feature.

- The open design attitude has been put in evidence through the survey of designers about their subjective level of tolerance, expectation and the objective observation of input's complexity
- The state of Flow has been put in evidence by the observation of the necessity of conjunction of Intuition, feedback speed and size of input informations in balanced proportion by experimenting design condition with various balance of those parameters.
- The Context encoding have been measured through an indirect index that relate the quantity of information provided in the input in order to get closer to the expected result. The remaining information that is not provided is implicit to the tool.

The observation of those parameters and feature is a challenge since its user dependant. Ideally, physiological measurement tools should be used to get objective data on the reach of the sate of Flow or the characteristic of the designer's attitude. Unfortunately, actual brain science and resources of this research don't allow such experimental system. Nonetheless, those experiment are good enough to show the importance of each of this features. There conjunction to create CR is still a too rare case to be studied directly but the advent of A.I. and V.R. will create the condition tools that contain more context encoding and GUI/IDE more intuitive and thus more prone to favouring reach of state of flow even for designers with weak programming and digital background. Finally, the design attitude is mainly a question of teaching design to the future generation. It advocate to a better understanding of the strength an weakness of computation in order to use it as the "tool want to be". To paraphrases the quote of Louis Khan that ask to a brick what it wants to be and the brick answer:"arch". The computer too, as a tool, want to be used in some way and the CR is the "arch" of the computer in a new paradigm of design where the design becomes the design of the process itself. Rare case of of possible case of CR presented in this thesis were mainly due to obviously a strongly oriented design attitude of designer and the advanced digital skills that allows to compensate the lack of intuitiveness and of encoded context. Indeed, through their skill, designer can pass out the threshold of the practice flow and encode themselves their own context. Unfortunately, this context is too specialised and personal since it requires great effort to be generalised to other designer. Above that, this encoding is specific to a problem and the vision of this problem thus it can't be transferable. Despite the code and the principles are explained, they are not open enough to generalisation in other design process. It can be source of inspiration to design new digital design process that will provide solution for other problems but they are not solidified computable knowledge that can be linked as a library in code.

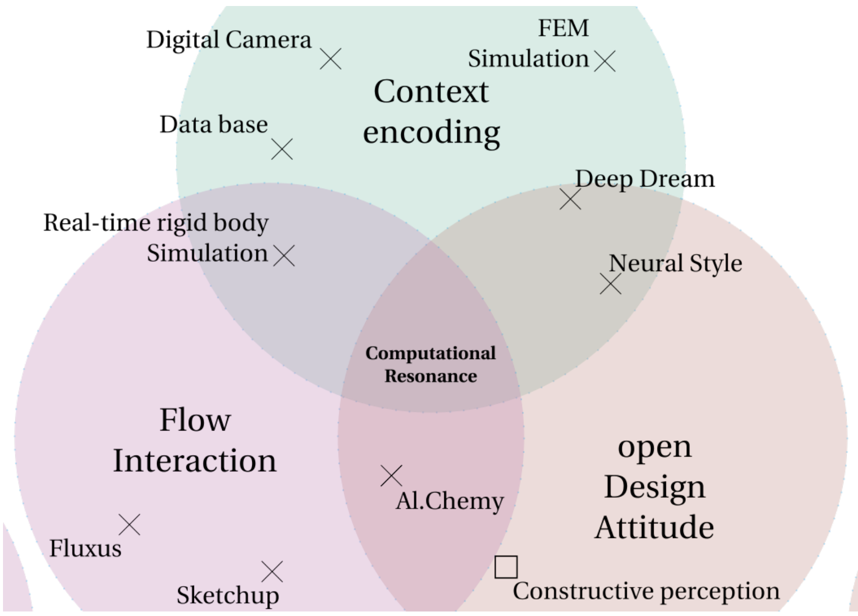


FIGURE 6.2: Specific attitude, suitable flow and encoding of the context are necessary in order to reach a resonance frequencies of Compute-Designer oscillation in case where the designer is not an expert in computation

6.2 Stakes of environment recognition

Initially, the goal and philosophy of the side project of automation of context encoding through ConvNet and photogrammetry is aiming to provide a tool component for digital design processes. This component was strategically a component for starting digital design process : creating the substrate model. This creation of substrate model is aiming to provide the option to format the data in such a way that other algorithmic tools can handle modification of the substrate model through new entry point. Above that, it allows designers to stock encoding of context in order to reuse them in digital design process as components, reference, data or any other mean that designer aiming for CR may invent for creating a tool context that help them to take design decision. Part of invention are always based on the copy and rearrangement of real experience. The effective copy into a computational format enable computational manipulation more easily. The initial project is limited to reconstruct through 3D modelling command the geometry captured from real environment. Nonetheless, the general principle is to use the ConvNet system to extract computational knowledge from reality without having to create a scientific model and implement it in the form of an algorithm for a later "toolifiaction" in a design process. We can imagine encoding through a video the reaction of bamboo stick to torsion through the processing of a video showing example of torsion to a ConvNet system. At another design scale, topological relations of room of architecture masterpieces can be extracted from a set of photographic pictures from the project site processed by a ConvNet. The use of ConvNet is quite new in Computer science and completely missing in algorithmic design field. Research on the encoding benefit of those systems for architecture algorithmic design can enhance the possibilities for designers to elaborate computational design process that relies on the conjugation of scientific method of tamed problem solving in order to get some feedback on the subsequent impact of their design choice.

6.2.1 Human as a link between natural and artificial

To advocate to the ideological criticism that may think that such research aim to put the designers out of the design loop and discharge their responsibility to computer tools. I'll advocate that on the contrary, this approach try to put designers at the center of their design environment in order to let them organize design process as conductor of an orchestra of method, digital tool, heuristics, scientific knowledge and site data that have been digitalized in their computer system. If the resonance frequencies of Computer-Designer oscillation can be reached for designers, it place the individual in link position between the full artificiality of computer and scientific knowledge to the natural environment where, in the end, all products of architecture designer should sustain itself. To achieve that, it need to be designed with the best comprehension of all the problems that interwoven the physical existence of thing in our natural context.

Digital tools for designer have been designed in the infancy of computer science and they are meant to be use for a practice that exist since the man is an *homo faber*. No doubt that the adaptation may take a while but ignoring that tools are shaping us too is missing an opportunity to improve the design as a science.

6.3 Conclusion

To conclude this thesis, it is necessary to remind that *resonance frequencies of Computer-Designer oscillation* is still a very prospective concept since documented design process that support this hypothesis is still seldom. Nonetheless, design science, CT, anthropological studies, cybernetic theories and other background research seems to converge on the idea that a utilisation of the computer tool that connect designer intellect on equal relationship at the expense to a pure top-down fashion is reasonably an expectable lead. Far from the sci-fi idea of a designer that discuss with an I.A. like HAL[1] as a design collaborator or an almighty slave that "checkmate in, say, 38484 moves"[85] the wicked design problem. A relationship with A.I. for design can be "designed" if a deep understanding of our and their intelligence is enlighten through further studies. This thesis propose a central concept of *resonance frequencies of Computer-Designer oscillation* as a guideline for such design and propose a theory of its working in a design process through the analysis of the relation between the designer and his digital tools. Flow, design attitude and context encoding are the three are that seems to be prominent in this relation. Nonetheless, the system is open to modification. Addition of other requirement that may encompass aspect of the 9 evaluations criteria that may have been neglected during the tool analysis or improvement of observation methodologies seems to be necessary to reinforce or disprove that theory product of our current computational paradigm.

[1]HAL is the AI in Space Odyssey, a movie of S.Kubrick

Appendix A

Component of digital tools

When the term "digital tool" is mentioned, layman may imagine a software such as Microsoft Word or Excel. This Appendix purpose is to modify this image of digital tool. Digital tool is essentially the computation. And the root of all possible software is programming language. But softwares and libraries exist to save labour of writing code in programming language. They are precisely helping in performing task as a tool do. Softwares, libraries are all tools to control tools that control other tools and so on, in order to control the hardware in the end. This recursion of tools is common when a computer is used. This Appendix describe the tools component by explaining basic of computer science : programming language, software architecture and computation principle.

A.1 Software and Tools presentation

To illustrate this vision of the digital tool and his different manifestation, the Appendix will present two workshop that have been conducted under the direction of the author. They are example of what is a digital tool separated form his framework context in order to create a new one and how digital tool are built.

A.1.1 Tool used in Workshop and Experiment

A.1.1.1 Belgrano University Workshop

Belgrano The workshop held in Belgrano University in Buenos Aires aims to teach to student a tool chain to produce easily shape that can be fabricated with laser cutter. The aim of this tool chain was to provide a easy means to produce model made by slice of wood assembled with simple junction. The tool chain was composed by Processing, Blender and 123D Make. In each software only few part have been explain. Each part that present feedback that is the result of an input message is considered as a tool. For instance the Solidify Modifier in Blender was one tool. It changes one surface into a volume. This volume is a thicker version of the surface. The purpose of this tool is mainly to adapt the 3D model to the need of 123D Make. Indeed, the 3D geometry needed as input of this software is different than the output produced by the processing script provided in the workshop. This tool allows the user to change the thickness of the initial shape. The user provides a simple input message, just a number that indicates the thickness in millimetres they want to give to the shape. The feedback is a visual feedback of the input 3D model modified by the tools accorded to the parameters set by the user. We can see through this description that the tool is not the software itself but the function in the software that provide a meaningful feedback for the design. Through import and export format of file format, the user can move from tool to tool. He is not bounded to the limit of one software and the set of proposed tools. Doing everything in the same software can be easier because the user doesn't need to process file format import and export steps. These steps are not considered as design tools since the feedback doesn't show a modification of the substrate model in a meaningful way for the design process.

A.1.1.2 CAADRIA 2014 Workshop

The aim of this workshop was to teach to the student to create their own design tool with Fluxus live codding environment. The workshop was held at Kyoto Institute of Technology in May 2014 before the CAADRIA Symposium. The brief of the project was to collectively build tools to design a building with fluxus as environment to produce the output. Each participant had to produce one tool that design one part of the project. The programs they design can be compare to a BIM component of CAD software. When the program is called, it draws an architectural component in the 3D space of Fluxus according the parameters provided by the input message. The final product of this workshop was a video that show in a live codding fashion the tool produced by each student. This video can be seen at this url : https://vimeo.com/95402815 on the date of 7th February 2017

A.2 Software as reference

A.2.1 Al.chemy

Al.chemy (Figure A.1) is a rare case of purely creative oriented set of tools. Open source software created to focuses on using experimental interaction in order to create new ideas. This software gather various tool that are mainly controlled through a graphic tablet. Unlike Sketchbook Pro which try to mimic the set of real tool that can be used on a sketchbook : markers, brush, pencil, etc; Al.chemy offer tools that modify the input. For example, the symmetry tool reproduce the input draw symmetrically to it's original position. Other tools propose such modification of the draw input in order to produce perturbation of the drawing habits. The principle of this set of tools is to trigger pareidolia of the user through the feedback. It could help in creative drawing process since the drawer tries to interpret shapes and based on these interpretations his drawing intention maturates. Al.chemy can be found at this url : http://al.chemy.org on the date of 7th February 2017

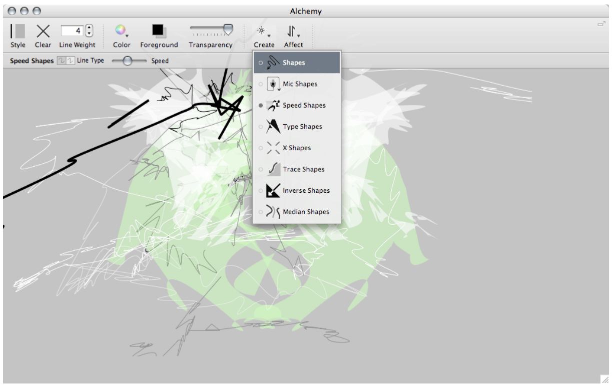


Figure A.1: The Al.chemy is oriented on experimental tool for creative drawing

A.2.2 GIMP

GIMP GIMP Figure A.2 is a raster graphic editor. It helps user to manipulate matrix of pixels with a set of convenient tools. They are all based on the modification of pixels. The type of tool can be divided in three categories : selector, filter and modifier.

A.2.3 Blender

Blender (Figure A.3) is a 3D modelling open source software. Due to open source development, this software contains tools that belongs to various environment : compositing,

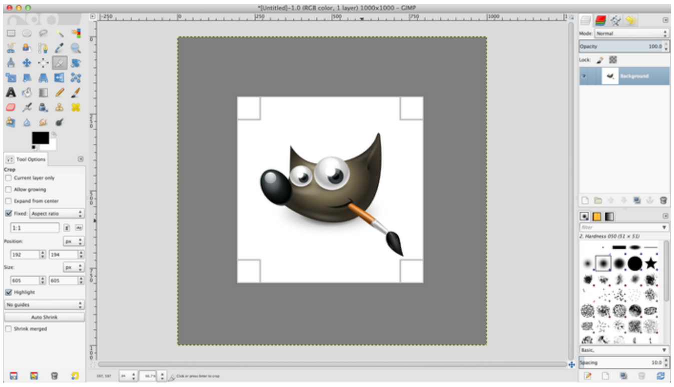


Figure A.2: The GIMP user interface allow to create and edit raster image.

animation, texture painting, etc. Natively, Blender handle mesh geometry in openGL. It doesn't allow the same precision and operations that SCG or BRep can perform. Nonetheless, some operation such as boolean operation are possible and other are unique to blender. Blender can be found at this url : https://www.blender.org on the date of 7th February 2017

Figure A.3: The Blender software is initially designed for CG movie production.

A.2.4 123D Make

123D Make (Figure A.4) is a software from Autodesk. This software is specialised in the transformation of mesh into blue print for digital fabrication process based on laser cutting and 3D printing technologies. The tools provided in this software flatten down geometry in planar separate part that can be assembled manually. For instance, it can performs an automatic slicing of the mesh. Those slice can be assembled by inserting parts in each other. The inserting slot are automatically drawn on the slice by an algorithm of the program. Then parts are organised in a optimal layout on a rectangular space. This format can be directly send to CNC machine for cutting. 123D Make can be found at this url : http://www.123dapp.com/make on the date of 7th February 2017

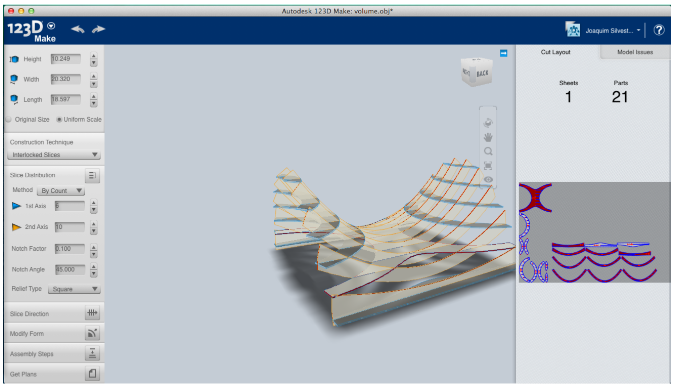


FIGURE A.4: A perspective view of the slice assembled and on the right the plan view of all the slice

A.2.5 Fluxus

Fluxus Figure A.5 is a live codding environment based on Racket language. Live codding environment favour fast compilation time and multimedia I/O management. For this reason, the code is displayed above the visual output. I/O are processed in real-time in order to provide a flux of content to the audience. Live codding practice is originally a form of performance that happen in underground party. The Racket is a multi-paradigm language, nonetheless his functional paradigm language feature makes it uncommon compare to industry standards. The functional approach of language allow to make recursion in the call of functions. Above that, language based on this paradigm handle flux of data instead of states. A flux of data can be assigned to a variable easily. Fluxus as a language is a tool to manipulate data in order to create program. The core part of the language, its grammar, is the tool but other tools as library or function handling data by adding a knowledge from a specific domain field, for example, light simulation for pictorial representation. Fluxus can be found at this url : http://www.pawfal.org/fluxus/ on the date of 7th February 2017

A.2.6 Processing2

Processing Figure A.6 is, like Fluxus, a programming tool and a IDE. It has been started by Casey Reas and Ben Fry in 2001 while they were student of John Maeda in the Aesthetics and Computation research group. This origin explain the digital art orientation of this software. Indeed, this software is used by many artist because it is easy

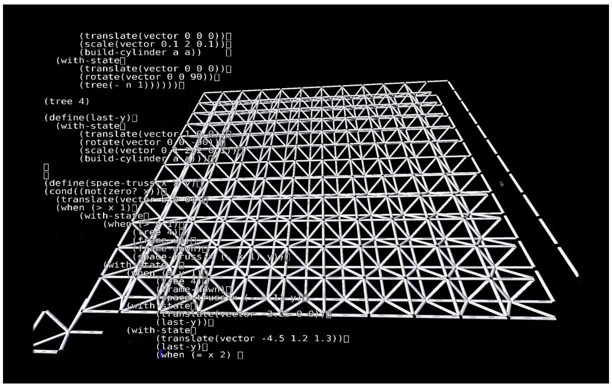


FIGURE A.5: Fluxus have a minimalist GUI, all the commands are done through keyboard shortcuts

for beginners. The philosophy of Processing is to propose a simple programming language bounded to a clear interface with only two buttons : play and stop. Compare to an IDE like Xcode that can be use with various language for industries' standard software development, Processing provide an interface that reduce the risk of being lost. In XCode, the menus and buttons are quite invasive and beginners are often encountering errors of linking graphical libraries at the beginning rather than feedback about how they write their code. Processing propose various libraries to expand core abilities of the system. Opening 3D model file, sending message to other program through OSC protocol and other abilities can be added to Processing through libraries. They are pack of tools that are related to a similar data handling method and related field of specialisation and knowledge. Processing can be found in the third major version at this url : https://processing.org on the date of 7th February 2017

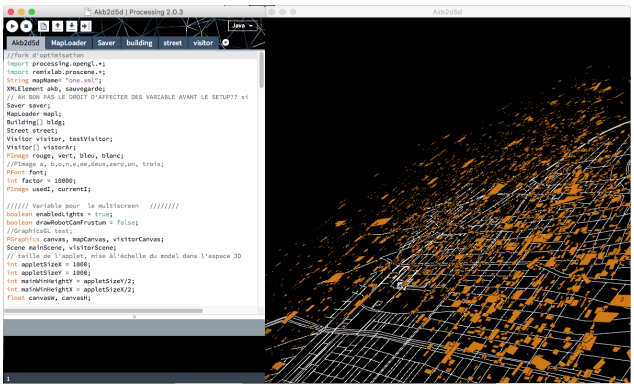


FIGURE A.6: Unlike Fluxus, the text editor of the IDE is separated from the graphic output windows

IGeo library

This library created by Satoru Sugihara enhance the 3D model manipulation in processing. 3D model of surface constructed with iGeo commands follows BRep modelling convention(Figure A.7). Other functionalities such as multi agent algorithms, physics Simulation and panelization make the library popular in architecture design school. Unconventional and complex shapes can be easily generated with tools based on these functionalities. iGeo library can be found at this url : http://igeo.jp on the date of 7th February 2017

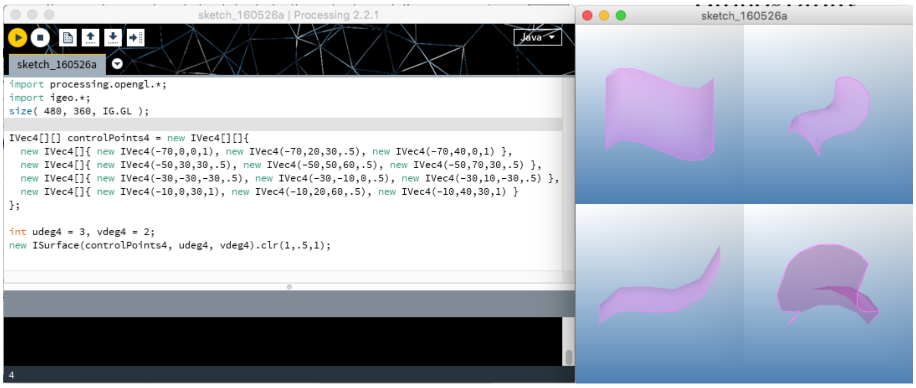


FIGURE A.7: This shape can be parametrically modified through the variable of the calling function

A.2.7 Photoscan

PhotoScan Figure A.8 is a commercial photogrametry software. It reconstruct a textured 3D mesh from a series of pictures. One specificity of the software is to be very robust. It can handle pictures even without the EXIF file. EXIF files are important because they provide the informations about the camera used to take the picture. Geometry of the lens and physical characteristic of the camera are useful to project back feature point during the bundle adjustment phase. It allows PhotoScan to process image rendered from a 3D modelling software. Tools like MicMac can't proceed this kind of images unless the parameters of the virtual camera are provided to the system. PhotoScan software can be found at this url : http://www.agisoft.com on the date of 7th February 2017

A.2.8 MicMac

MicMac is an open source platform for photogrammetry. Unlike PhotoScan, MicMac is a command line based interface. Through the Terminal, user type command line to launch

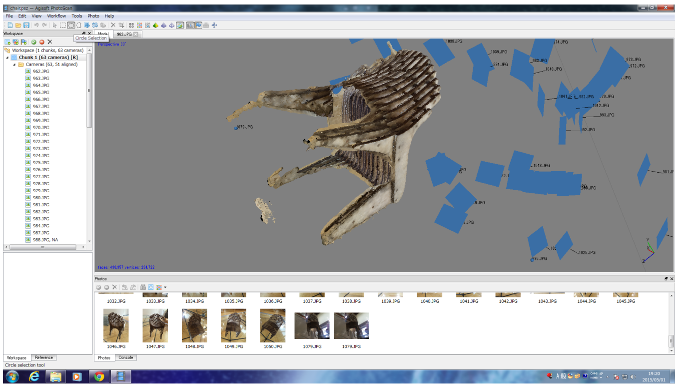


FIGURE A.8: Photo scan transforms the series of pictures of the same object in the lower part of the picture into a 3D model. Camera orientation and position are represented by blue rectangle in the 3D view.

the different tools of MicMac. To look the output file produced by MicMac, MeshLab is advised. MicMac is in fact a platform that articulates different tools previously created: Pastis, Apero, Tapioca, Tapas and C3DC. All this tool process one step of the general process of photogrametry. The open source aspect allows upgrade of the tools and addition of new tools that use the same data management process. For instance, one additional tool developed recently by Adeline Manuel [86] called Aioli allows to spread annotation made on one picture to other pictures that present the same picture elements seen from other point of view.

A.2.9 OpenCV

Open CV is a library for computer vision. Various tool are available and can't be listed here. Nonetheless, knowing those elemental tools are important to understand ConvNet, photogrammetry and every process based on image processing. Here some functions mentioned in the thesis are explained. Those function

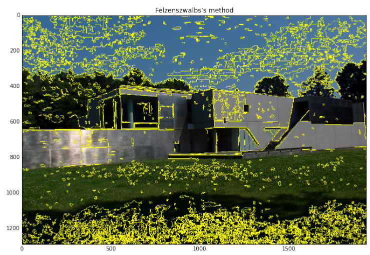
Harris Corner

Harris Corner and other type or features such as FAST or SUSAN are key elements for photogrammetry. As the recognition of singular objects allows to guess that 2 pictures are taken from the same scene, features are the singular points that are use during the bundle adjustment. This feature have an vector identity that is invariable to rotation, translation and illumination. If the identity of two points of two different pictures are similar, the 2 pictures frame the same object. As we see in Figure A.9, the point appears where the images contains details. Long and untextured area don't contain Harris Corner. For area and lines other detector exist. It will be too long to list them here but the most important is to keep in mind that their basis for detection is mathematical analysis and that the human eye only that project a meaning on this points, area or line. Sometimes the line detected can match with the frontier of an object but it can detect the boundary of an element in the texture pattern. There is not semantic meaning to the detected part.

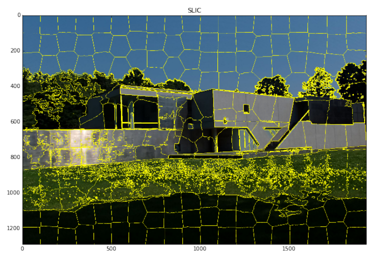
FIGURE A.9: Harris & Stephens detector applied on a CG, we can notice that corner appear in area with details

Segmentation algorithms

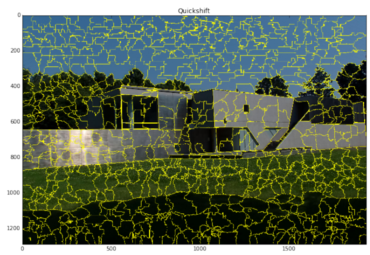
In computer vision research, there is various algorithms to segment an image base on various criteria. These method are using mathematical analysis of images in order to detect variation of colors with various threshold in order to gather within the same segment pixels that present similar color values. It may be enough to detect mono color object on a high contrast background but there is no understanding of "object" as they make sense for human. We can see in the Figure A.10 different algorithms of segmentation all based on a similar mathematical analysis strategy.



(A) Felzenszwalb algorithm : large area are selected with some discrepancy with intuitive expectation



(B) SLIC algorithm : All area present average size but still, very small segments are creating noise in the output



(C) Quickshift algorithm : seems to be a compromise between SLIC and Felzenszwalb

FIGURE A.10: Three tools that perform the same task: segmentation of the whole picture.

GraphCut

As seen in the previous tool, automatic segmentation is not yet perfectly autonomous. Nonetheless, with small valuable clue provided by the user, GraphCut algorithm can perform very clean and neat segment that match with user expectation. The draw back is that only one segment at the time can be created and it is not exempt of flaws as we can see in Figure A.11. One small rear spike of the shape is not inside the segment.

(A) Original image before being segmented with GraphCut

(B) Result after the algorithm processed the image.

Figure A.11: An interactive tool to segment image

A.2.10 MeshLab

MeshLab (Figure A.12) is a tool for visualisation of 3D points. The tool can process low level 3D models such as point cloud or mesh. It present the model with basic camera movement interaction to inspect the model. There is no design action that can be performed on the mesh.

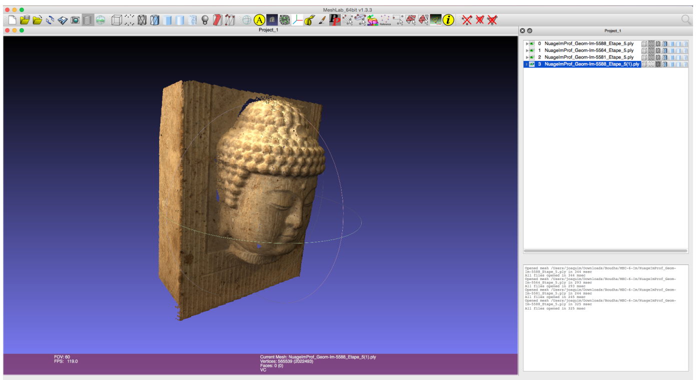


Figure A.12: Meshlab allow few operation such as hole filing

A.2.11 Rhino

Rhino is a BRep 3D modelling surface. This software is very popular in architecture, boat design, object design, cloth design, jewellery design... Beside his representation mod that allow different kind of control and manipulation on 3D model, Rhino is very precise and present some exclusive tools. This tools are only available because of the BRep geometric model of representation. Additionally to this core tools, plug-ins provide other set of tool related to specific topic. For instance, Kangaroo provide rigid and soft bodies physical simulation. Another asset of Rhino is it's scripting tool that allow to use all the tools of Rhino through programming language. It allows automation and by this way, creation of new tools in the Rhino tool box.

A.2.11.1 Grasshopper

Grasshopper is a popular plug-ins that is used to do visual programming. By connecting boxes that represent tools, user can design a tool chain that will produce a shape from input data. Each box have input and output slot. The input data should be chosen according to the type of data the tool can handle. Tools have specific input and output, for instance one tool can take a 3D point and a real number 'x' as input and provide a sphere of 'x' rayon. Through long chain of tools, User can create a full new tool.

A.2.12 Geomagic XOS

Geomagic XOS is reverse engineering remodelling software. It provides a set of tool to analyse mesh surface and determine what kind of surface they are : developing surface, free form surface, parametric surface, flat surface ect. With the intervention of the operator, the whole surface of the volume is registered in various type of surface. Some of the surfaces can be detected automatically but the operator intervention is necessary to check and correct the generated result. Then, the operator complete the registration manually by selection method similar to the magic wand in GIMP.

A.2.13 Torch7/ neural network framework

Torch is a scientific computing framework that can be controlled by Lua based script language. Basically it's a matrix calculation system like MatLab. It handles matrix of numbers and multiply them with numbers of another matrix. Since ConvNet are made of layers of neurons that have bias and weight, they can easily process through matrix. On matrix represent all the weight of all the neurons in one layer. Torch7 is enriched of libraries that are called "luarocks". This library extend and specialise Torch domain of application.

Appendix B

Convolutional Neural network in Artificial Intelligence

Artificial Intelligence is a field promising for digital design tool. Tools that use A.I. component enhance user's ability with more than the mere automation usually offered by computer. Their added value represent savings in intellectual work for the user. This part will present different field of A.I. then the field of Machine learning will be explained in order to explain the stakes of Deep learning with ConvNEt.

B.1 What is real and artificial intelligence

Artificial intelligence is a branch of computer science that embed a part of philosophy since it uses a notion that is not clearly defined. What is intelligence ? animals or plants in their reaction are they intelligent? Can we measure intelligence? IQ test are they taking in consideration all the aspects of intelligent thinking? Lots of answer that have not been decided. Usually, artificial intelligence is divided with the strong and weak. The strong is performing what the human intelligence does and it is oriented to general-purpose task. This is typically what can be seen in SciFi films about androids and robot. The weak one is what the human brain should do and is oriented on special purpose. In this research the author focused on the weak artificial intelligence. It concerns only algorithms that do some simple task with a form of intelligence. A task is a valid AI challenge if within this task a small autonomy required.

B.2 The A.I. tools

In A.I. field tools such as

- logic paradigm programming language
- artificial neurons
- Markov chain
- Baysian network
- Machine learning
- CRF
- ...

These tools are used because they can handle situation with some autonomy. One recognizable feature of intelligent system is the ability to take a binding decision. For instance, the probability of Markov Chain or CRF can provide answer based on probability when all the information needed to take the decision can't be known.

Until now, algorithmic design practice didn't make much use of A.I tools in their project. The designers were focused on the use of algorithm to solve or to simulate. Another goal was to take advantage of the rapidity and efficiency of the automation in computing to achieve a flow of interaction in the design process for emergent behaviour. And finally, enhance the data work flow that integrates CNC machines.

B.3 Machine learning

In the category of A.I. system, the thesis will focus on the Machine Learning process. On the basis of training data, those system extract information that will help the system to predict the answer on new cases. This process is a kind of inference based on statistic. Those statistics are created from training data. Information is recorded on a n-dimension vector format. Classic algorithms used in machine learning such as K-means clustering find correlations betweens those vectors during the training phase. Learning algorithms are divided in two categories :

- Supervised
- Unsupervised

Supervised learning algorithms learn form the data a feature specified by the programmer while unsupervised found this feature and learn it. For this reason, unsupervised tools are often related to data mining. Data mining is about founding correlation between data. This correlation is used later as feature for a supervised learning. To summarize we could say that unsupervised machine learning is data mining coupled with supervised machine learning. Based on simple vector data, those algorithms can only learn and found very basic correlation. They can learn classification or segmentation of data but they can't learn to see. Various algorithm that record the "knowledge" to learn from the data.

- linear regression
- regression loss functions
- stochastic gradient descent
- linear classification
- logistic regression
- sigmoid
- ...

Basically those algorithm are mere functions called activation function. They can be used to define a predictor. That predictor analyse the input data and make a prediction. The predictor can make prediction such as:

- Classification
- Regression
- Ranking
- Structured prediction

To resume it can learn to classify an item but can't answer to an open question. The training aims to define from all the examples a kind of average predictor that can provide the same answer than the one stocked in the training data. For instance, to teach ranking to a machine learning system, Input can be an unordered list: L, C, B, D. And the output, the expected correct answer : B, C, D, L. Training is followed by a testing phase. Training and test data set are the same type of data. They are set of input and output, the input is the question and the output is the expected answer.

B.4 Artificial Neurons

Machine learning principle can be model with artificial neuron. A sole neuron can make simple calculation from its input and send the result through its multiple output. The artificial neuron can perform more complex calculation. For instance, inputs can be weighted differently and an activation function such as the logistic regression can be implemented in the neuron. Usually, neurons work as network and can perform task. To designing the network, general structure of the network need to be made by selecting a number of neuron and connecting their input and output together. Then by putting weight on each inputs of neurons and deciding activation function for each neuron, a whole neural network is set-up. Input of a neuron and weight of the neuron can be seen as a vector. They are called weight vector and feature vector. Each entry are one dimension and the weight of the input indicates the importance of the dimension in the final calculation of the prediction. The most basic activation function set a threshold to trigger a binary answer instead of a real number. By selecting the activation function, it define partially the role and position in the network. The neural network described here doesn't learn. All the knowledge encoded in the network is manually encoded by a programmer.

B.5 Learning with Artificial Neurons

To automatize the learning, the structure of the network is standardized by a layer system. Each layer contains neurones and each neurons of each layer are connected with the previous and next. Usually, all the neurons of one layer have the same activation function. Such format of neural network is called shallow networks. It contains one layer of hidden neurons, the neurons of this layer are connected to input and output layers. During learning process, the weight and bias of those hidden neurons can be modified. The setting of the weight and bias is a way to encode the feature detector into the network. To encode the feature automatically on the basis of given examples, the back-propagation algorithm is used. The principle of this algorithm is to adjust the weight and bias in order to make the output matching with the expected result. For instance, if according to the training data set, the expected result when the sequence "6, 7" is given should be 42. But, the output of the network is actually 62, the back-propagation algorithm change the weight value in order to reduce the output to 42. This very simple example of multiplication shows the basic principle for one neuron. In practice, the back-propagation algorithm need to do an average between different expected result and effective output. In deed, with lots of neurons, there is a lot of possible setting. To avoid over specialisation of the detector, training are done by batch. By this method, the back-propagation provide an average setting of neurons that satisfy various set of input output training data.

B.6 Deepen the network with hidden layers

Now that the general principle is explained with shallow neural net, the principle can be extended to deep neural network. Deep neural network contain more than one hidden layer. therefore, multiple hidden layer allow to set different activation function for each layer. All the art of Deep neural network architecture is to select the proper sequence of layer in order to enable the system to learn the desired factor. The backside of deep neural network is the difficulty to back propagate the error to the first layers. Indeed, last layer can be easily tweaked to get the expected result, it follows that the error delta that need to be propagate is quickly corrected right from the beginning of the back-propagation process. Nothing left to be corrected after few layers modifications. Depth of neural net provide the space for other type of layers. These layers don't have all learning ability, they exist sometimes just to re-arrange the data. It's especially useful for preprocessing the data with a specific feature detector that doesn't need to evolve. Generally, this is well known functions that are performing specialised tasks. Other layers are less specialised. Their specialisation is created through learning. For instance, layer of activation function such as Tanh or Sigmoid, have variable weight that are adjusted during the training process. In emergent way this layer becomes specialised. This specialisation is called feature detector and can't be imagine as precise and stand alone specialisation. It's usually a specialisation in a hierarchy of sub-specialisation and meta-specialisation that are able together to perform a specialised intelligent task.

B.7 ConvNet for images

Until know, explanation where based on network that take a n-dimension vector as input and provide a p-dimension vector as output. With n the number of feature used to describe the input and p the number of possible output. Let's expand the network with a larger input format that fit to handle images in ConvNet. In Torch (Appendix A), instead of managing vectors the system work with multi dimension matrix. These matrix are called tensor and can have various format such as :

$$\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}$$

for a 3×3 tensor or even

$$\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}$$

for a 3×3×2 tensor

Usually, color pictures are h×w×3 tensor. For instance, a 800×600 pictures become a 800×600×3 tensor. 3 matrix are needed to record the RGB values. According to the Torch documentation, there is no limitation to the size of a tensor. With those tensor, layers of neuron handle more inputs and they reveal their potential for image recognition. For instance, the group of layers that gives its name to ConvNet, Convolution layers, are a key component for feature detection. Layers in ConvNet can be seen as multi-parameter functions with weight on the parameters to activate or inactivate them according to their utility in the construction of a detector. Weight can serve to turn layer on or off for some neurons and not other. For instance, the input of the upper part in a picture may be connected to set of neurons prone to detect bird part. The same feature detector will have negative weight for the

- *Convolution layers*: They gather inputs that comes from locally close neurons. By this way, the calculation operated on a group of input related to the same portion of the picture. It allows to recognize local features but others layers such as Spatial Pyramid can recognize features that are at the global level of the picture.
- *Simple layers*: They adapt the Tensor to a proper type of value. For example, the Abs layer give as output the absolute value of the input.
- *table layers*: They arrange the layout of tensor. They don't perform operation but change the structure of the tensor. For example, the JoinTable takes two tensors and join them into one. It can be use to join two pictures in one when they are in a tensor format.

- Simple layer such as Linear, mean, Max, Reshape, Exp, Sqrt
- Transfer functions layer : HardTanh, Tanh, HardShrink, SoftMax, Log Sigmoid, ReLU
- Table layer
- Convolution layers

B.7.1 Training

B.7.2 ConvNet for identification

The sum of abstract feature detector allow the recognition of a concrete feature in the image. Meanwhile that abstract features can't be understood by the human observation, the sum of these feature are recognizable by humans. If we make a parallel with human vision process, to recognise a face, we first look for eyes, if their is one or two eyes, we look for a mouth close to the eyes. If between the eye and the mouth we can recognize a nose, we are sure it's a human face that is represented. This description a bit naive, shows a recognition process. For a computer, it's similar except that instead to have sub-recognition task that have for aim an object or a part of an object, the ConvNet will look for a pattern of group of neurons activated together. For instance if Layer 1, neurons 3, 4 , 5, 66, 100 and in Layer 2 neurons 10, 50 to 60, 77 are activated then Layer 3 have neurons 5, 8, 10 so the output will present a face. Despite those features are abstract and hardly decipherable at the individual level. As a group belonging to a layer and in comparison of other layers, a hierarchy can be recognised as we are going to see in the next section.

B.7.3 Deconvolution

Researchers in ConvNet were curious how the knowledge is organised in the network after the training. How the network store and organise the recognition sub-task in order to perform the task with such high rate of success. Understand the organisation offer insight on another way to "see". Indeed, in competition of computer vision challenge, before the wave of the ConvNet trend break-out, most of the detector were made manually. They where invented by researcher through mathematical analysis and reasoning. So they can be explain and reuse as component in other detection system. If the generated detector of a ConvNet could be understood. It can be extracted and reuse with other detector. An assembly of detector can be designed like we assemble lego block. The tool build with this top-down fashion can be more reliable and provide more control option to the final user. Few manually crafted detector : The corner detector type are key component of photogrammetry system. If new and reliable detector can be created, they may be reuse in other task than the sole recognition.

Unfortunately, it can't be done by the analysis of the weight and bias in the trained network. A mediated way is needed because, indeed, our vision system is not trained to look at networks of neurons with thousand of parameters. Human vision system is more efficient to see structures, flow pattern, color heating. Tools to visualise and decipher this knowledge encoding have been tried and they help often for teaching the working principle of ConvNet. Visualisation attempt have been tried such as the visualisation of the network flow when it's in functioning. This *side* view is like a longitudinal section of the system. It shows the inside of the a living system and the observer can have insight of pattern by experimenting with it. Other systems offer an abstract images of the different filter. This is more like a transversal section of the system when it is off-line. There is no point to see this representation dynamically since it represent the information of the training data set that have been solidified in a computable form of knowledge. But these representation doesn't help to decipher completely the working of ConvNet. This field is often considered as craft because the precise working of the system is unknown. The name itself, "Convolution", indicate the complicated and obfuscated aspect of the system. In order to get better understanding, a visualisation technique was created. It is called "Deconvolution" in a reference to a process that de-complexify the encoding. The main Idea of the deconvolution is to gradually modify the image to see which neurons are activated in the network. These neurons belongs to a abstract feature description and the gradual change tend to produce shapes that trigger an optimal reaction of the group of neurons that belong to a feature. Since feature are interrelated and neurons alone don't have much explicative values, the group of neurons chosen to measure the activation rate that will drive the image alteration are a full layer of neurons. When using deconvolution tool, an initial image and a layer is specified. By looking the results of experiments with the same image but different triggering layer. The human brain can distinguish pattern and implicit hierarchy in the type of shapes that are "stimulating" first layer and those that stimulate last one.

This program is presented as a tool for images production in the appendix A and have been use to conduct experiment on design tool in the Chapter 3.

B.7.4 identification and localisation.

The previous part explained that features trigged during a detection can be detected with ConvNet visualisation tools. ConvNet that perform detection in natural image can detect more than one elements in a image. For instance if this picture B.1 is submitted to a ConvNet is trained to recognize car, truck and motorcycle it will detect both car and truck because they are in the same picture. Usually it detects different elements and rank them through a comparison of detector activation strength. But the result is unpredictable and if this ConvNet is used in a self driving car it's safer to have guarantee output.

FIGURE B.1: Since the ConvNet class pictures in categories of object defined during the training, the system may detect more class the picture in more than one category

To solve this unpredictable behaviour, the problem submitted to the ConvNet should be re-defined. What is on the picture and where it is located in the picture is a correct definition of the problem. To locate object on the picture various techniques are possible. Firstly, the picture can be randomly segmented in rectangle and each part are resubmitted to the ConvNet until it detect the object it had previously detected in the larger picture. Then with the coordinate of the picture segment an average localisation of the object can be given within the image coordinates. A similar system based on the parsing of the image through a small window could produce even more precise result. The main flaw of this technique is the variation of size of the object in a picture. A very big car in the foreground and a very small human in the background may not be detected within the same parsing window. Usually, big object will be found in the picture but the too small parsing window will fail to locate it.

This problem will show in the next section the advantages to associate automatically made detector with more mathematically formal detector crafted *manually* by an expert.

B.7.5 ConvNet for segmentation

Segmentation is about identification since when segment are made, they are limit between something and something else. There is a need of criteria of identification of at least two different thing to trace a frontier between. But, as we seen previously, localisation is vague and drawing a frontier between two identified objects is hard if the localisation system not precise enough. To precisely delineate the frontier of an object, the parsing window should be the smallest possible. In computer image, the smallest unit is the pixel. Nonetheless, scanning at the pixel level the complete image is designed to fail. The detection of objects is possible only at the size of a group of pixel and is greatly supported by the surrounding information in the picture. If the picture is taken in a sea environment, a group of pixels are more likely to be a fish than meal. If the fish look like a meal, if similar feature to the meal and the fish are detected, the other information around will help to solve the question. This correlation between feature detected happen often at the last layer of the ConvNet system: the fully connected layer.

To summarize, to detect and locate precisely by segmenting at the pixel level, the image need to be parse with the smallest possible window but if a too small windows is used, it may fail to detect the object in the windows. If the parsing windows is more than one pixel, it's geometry will impact on the precision of the segmentation line drawing. As we seen in the Deconvolution section, we have seen that with ConvNet working visualisation tool it is possible to trace back which small group of pixels have triggered the feature related to the detection. Even if the information of other group of pixels have triggered other features which participate to predict the good answer, their is a group of pixels, right at the input side of the pictures that if modified, the object will not be detected any more. This group of pixels is an abstract detail that recover the same importance as the detection of an eye in the recognition of a face for human mind. The problem is that small group of pixels belongs to an object or a category of object but it doesn't delineate the object itself. In a human mind recognition process, it's like recognizing "face" because "eyes" evidence have been found but being unable to segment that face from the background. Only few spot of face can be showed through the trace-back of the network. This loss in precision is due to the progressive reduction of the number of neurons per layer as we approach the output. Closer to the output, the number of neurons per layer scale down to the number of neurons in the output, the number of possible answer.

To compensate this loss in precision in localisation for the profit of identification researcher in scene parsing associate this system with a segmentation algorithm based on well known detector manually crafted. This segmentation is essentially based on geometric propriety of the images of the pictures. It look at values of pixels and their average variation to set frontier that follow group of pixel with approximatively the same pixels value. Those crafted segmentation tool exist with different algorithm, you can see in the tool section appendix A. This geometric segmentation help to extend the area of detection to a full segment. Thanks to its geometry following borders of objects in the image, all the pixel of segment belong to the same object. This principle is demonstrated in the work of Clement Farabet. [87] for more detail. In this work, the complete picture is classified into different segment. This method, called scene parsing can't work to segment and identify all existing object in all possible pictures. The number of type of object is limited by the number of output of the ConvNet. To succeed to classify all the pixels of the picture into categories, categories need to be large enough. It is possible to make categories specialised and a garbage category that will take all the pixels that don't belong to the specialised one.

B.8 Recursive Neural Network

ConvNet are mainly use for images but they can be used with any kind of data. The requirement are uniformity of the data format and the possibility to access to lots of sample of this data in this format. After that, this is just a question of formatting the tensor according to hardware limitation. Nonetheless, some formatting are more appropriate than other if features want to be learn. For instance, pictures are a type information that make sense in whole. If cut in band and streamed linearly, their meaning may be lost. By contrast, a sentence or a song make sense in its temporal dimension. It's because the subject is before the verb that it is the one who perform the action. If words like [John, Bill, kill] aren't proposed in sequence but always in one block, it's difficult to find by a learning process that the word coming before the verb is the subject. The ConvNet will never learn to make a correct a prediction if it's trained with a format not adjusted to its characteristic. For recognition system that work with data that implicitly use time to encode informations, the solution is to use RNN. Those network process sequence instead of group. It learns form the occurrences of repetitions such as sequence patterns. This architecture of Deep neural network shows its efficiency for text analysis or speech processing. As ConvNet, RNN can be turn upside down and be used in a "Deconvolution" mode. The process is a bit similar to ConvNet. On a base of a seed data, the RNN will generate data that are the most likely related to this seed. With a small string of character as input, if the RNN have been trained with poetry as data set, RNN can predict next character then the based on the previous string and the generated character it will generate another character and so on until generating a full original poem. Another advantages of RNN is that they can process vector of any size if they are presented in a stream of vector. If training data is too big to be processed in one block, the system will process it through a parsing window that will progress along the image. If the windows is big enough to catch features those features the Deep neural network will learn to retain important feature in the picture and correlate this feature with other in order to make the correct prediction.

B.9 Size and form of ConvNet

ConvNet is made possible by the advent of powerful graphic cards and optimised software like Torch7, made to use efficiently the hardware of these cards. For instance, the CUDA library give control of the calculation core present in NVidia graphic cards. This importance of the hardware reflect the greed in calculation resource of such system. For this reason, the input format need to be very close to the hardware input format. Efficiency can't be loose in back and forth data between graphic card memory and RAM or hard drive. For this reason size of the input images are limited by the graphic card memory. Since each pixels of an image are one input, the size of the input image depends of the size of the ConvNet. Large ConvNet with lots of input can process big pictures. Usually, the picture can be represented as a tensor $h \times \mathrm{w} \times 3$. For a web picture h is less than 1000 pixels and w under 2000. A ConvNet with approximatively 600000 input neurons can process most of the web images. On the other hand, ConvNet can be much or less deep. The deepness is the number of layer from the input to the output. For instance the bvlc googlenet have 15 layers when the MIT Places ConvNet have only 7.

Appendix C

3D capture technologies in Computer Science

The attempt to capture the environment in 3D is possible with various techniques. Those techniques depends on the type of 3D model expected. This Appendix will present different type of 3D modelisation and different technique of 3D capture.

C.1 Capture in 3D, translate the reality into a Geometric model

Capture in 3D is a generic term to designate the recording through sensors geometrical information of the environment. The data provided by the sensors are electrical signals that need to be processed with algorithms in order to be recorded in a compact and convenient format. We will see in the part about 3D format the question of the file format and paradigm used to establish the model to put data in form. Then, in the Enriched format of model section, the author will present informations that can be added as a layer of information on the geometrical information. Two cases will serve as examples. The first one will be about the collision map that is often used in video games environment. The second is the BIM format that is becoming the new standard for architecture building blue print formatting. Then a section will be dedicated to present diverse type of sensors that can be used acquire raw data. Device such as DSLR, Plenoptic Camera, ScanNect and LiDar will be presented. The algorithms that transform the raw data into proper 3D model will be presented through two examples : Sfm and SLAM.

C.1.1 Variety of 3D model establishment paradigm.

3D modelling software exist before digital capture system. Before being able to record data form the environment, 3D models were constructed manually. The user had to specify to specify through command how the shape should be. The way to specify the shape obey to a general paradigm. Indeed, even if a mathematical description is the base of all 3D model, various branch of the mathematics can be used to describe a model. Those ways to describe mathematically have impact on the precision of the 3D model. Depending the paradigm of the software, design tools it contains need to be use with different strategy. The following sections will present different paradigm of 3D depiction. There will be not presented extensively like in a Wikipedia article but the key elements necessary to understand the thesis in the main part. Above that, those descriptions don't relate the reality of the data structure we encounter in the output data produced by available software. Data structure in software take in consideration, colors, textures, layers, object group and other information that help organisation of 3D scene – arrangement of multiple shape in the same virtual space. Those paradigm can be graded from the most basic and homogeneous type to the most heterogeneous and refined data structure. By homogeneous data type, the author means that the general format of the data is an accumulation of similar component. An example of an homogeneous data format will be provided with voxel. [1] This scale of homogeneous to heterogeneous match to gradation between computer functioning logic to designer functioning logic. We call this scale : *machine to human scale.*

C.1.1.1 Depth Map

This format is almost not a 3D format, to be precise it's called a 2.5D format. This format is similar to 2D black and white image format except that by convention pixel value correspond to the depth value of point. This depth scale is relative to a reference point. This reference point is not indicated in the data format so the depth value are somewhat relative to each other instead of an accurate measurement of the depth. Depth maps C.1 using TIFF image encoding are limited in their precision. Indeed, the 128bit encoding of grey level limit the range of depth between the deepest point and the closest point to 128 level of depth. Since those value are integers, the degree of precision encounter the same weakness than Voxel modelling paradigm. Intermediary value between two level of depth can't be recorded in this format. Above that, this lack of precision, this representation is truncated. It can display only one face of the volume. And if the volume is hollow,

[1] Since implementation takes care of hardware optimisation concern, the real data structure may be more complicated than the given explanation. Given explanations are the theoretical principle of functioning

this representation fails to express this feature. Despite its numerous weakness, this file format is simple and easy to process. It can be very useful for getting an approximative topography of object that can't be fully modelled because they are too large. This data format is used to model mountain topography and Bump texture.

FIGURE C.1: We can see on this depth map various group of pixels with a similar level of grey, they correspond to the various depth plan. When the grey is lighter the plan is closer to the camera.

C.1.1.2 Voxel

Voxel C.2 with Point Cloud are the simplest way for a computer to store informations that can be seen as 3D shape by the User. Voxel are discrete 3D pixels that are arranged in a 3D grid. They hare the 3D equivalent of raster graphics. Since it's a discrete grid, voxels can be located with integer value. It ranks Voxel closest to the machine in the *machine to human scale*. Indeed, the 3D data format match with its memory storage model. Each memory slot is a point in the 3D grid. If the memory slot is 1, there is a voxel at the corresponding 3D grid location. Voxel depiction require lots memory space if compression algorithms aren't used to rationalise memory usage. As a depiction system it present few advantages. Boolean operation between shape are very easy to handle. Nonetheless, the general precision of voxel information is always limited to the grid resolution.

C.1.1.3 Point Cloud

Principle of Point Cloud C.3 are similar to Voxel. It stores point elements in a Cartesian Space. Their accumulation and their size create raster graphic images when looked from a virtual distant point. Other shared similarity is the absence of relation between elements. But contrary to Voxel system, Point Cloud can depict density of 3D point.

FIGURE C.2: Despite that popular game Minecraft isn't technically made using Voxel modeling paradigm, it is its best representation

More point's can be accumulated in an area than another without changing the overall shape schema. Point Cloud can be seen as little bit more complex since they use floating point coordinate to describe the location of the point in the Cartesian Space. Despite floating point precision is limited, it's a closer to human reality measurement. Precision is still limited to a resolution but it can varies more easily. In a manner of speaking, the Point Cloud relies on the intuition of human perception to convey shapes. The empty space between two 3D points in a Point Cloud System doesn't imply that they belongs to two different shapes. On the other side, discontinuity in the voxel grid implies a separation of shapes. That's why, a conversion from a point Cloud to Voxel depiction will generate wrong shape. By wrong shape, it's important to understand that the wrongness doesn't come from imprecision. That wrongness can be a topological wrongness. For PointCloud and Voxel scene, elements that are belonging to the same shape are not explicitly stored in a group. Voxel and Point Cloud doesn't have group so the 3D scene is a flatten data structure.

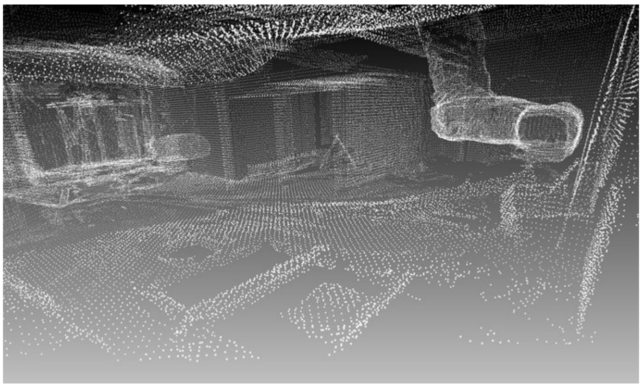


FIGURE C.3: This representation mode need high density of point to display shapes that can be readable

C.1.1.4 Mesh (Unstructured grid)

Mesh C.4 are 3D Point Cloud with connection between points. Those connections are vertices that forms triangles facets used to depict surface of object. Compare to Point Cloud and Voxel paradigms that encompass in their model establishment the ability to depict the shape content. Mesh paradigm depict hollow surfaces. These surface can be closed to give the illusion of solid volume but inside the mesh model, no information exist to indicate that empty space is enclosed or not by a mesh surface. On the other hand the boundaries is clearly modelled and intersection of surface can be precisely calculated. The position of points are still limited by floating point limitation of the coordinate system of the Cartesian Space but precision of the surface convolution can be precisely registered in the data format of the mesh. Triangles can be more or less small and dense depending area of the surface. It helps to increase precision in area where it's needed. For the area with simple surface geometry, large triangles can be used to save space in the computer memory. The memory storage of mesh are stripe of triangles. It allows to implicitly stores connectivity of points without the need to use two data structures: on for 3D points and one for connectivity between those points. For Mesh scene – that contains more than one surface, points that belongs to the same surface are gathered by group. We have a first degree of hierarchy in data structure. This paradigm is the one used in Open GL libraries. Those libraries are low level libraries that are generally used to drive hardware graphic card. This is due to an interaction of various factor such as policy of graphic cards constructor, trend in 3D programming, business and marketing of video games in the 1990.

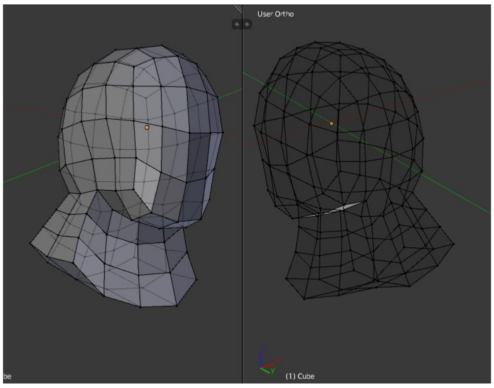


FIGURE C.4: Even without texturing and shading this representation is human-readable.

C.1.1.5 BRep

With the BRep C.5 the data format reach threshold of complexity. Like the previous model paradigm the simplest 3D data elements is the 3D point that can be stocked with 3 variables. Like the Mesh paradigm, those point can be put in group. But unlike the mesh model paradigm, there is a larger variety group. In BRep depiction, information about combination between elements, and geometric informations are stocked in different group. Geometric informations are not only coordinate of 3D points but it includes canonical curves supporting segment and canonical surfaces supporting face informations. Compare to mesh there is enrichment of topological information about element relations. The construction of BRep shapes can be described through group of points that from line and groupe of lines create surface. Each format of group handle elements with a specific method to produce a specific shape. This structure is close to OOP language structure: there is class that correspond to geometrical form category. And each object produced by those class correspond to a singular case of the category. The group structure of curve is the class of all possible curve. A group of three 3D points in this group makes one specific curve. If their is no hardware limitation, Mesh model paradigm can describe

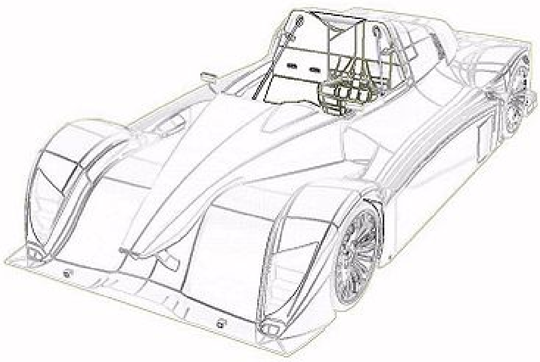


Figure C.5: This mode of representation fits to represent all objects made from developable surface.

all possible surface. The structure of the model scale linearly with increase of the size and level of detail of the mesh. On the other hand, BRep model structure scalability depends of the singularity of the model. If the shape to depict is close to canonical shape, the size of the model will not impact on the storage space since the level of detail is bounded by the fact that the shape can be reduced to short definition. If the shape depicted is far from canonical shape, lots of segment of canonical surface need to be stitched together to depict precisely the intended shape. Describing all possible surfaces with Brep is possible but it will be at a cost of a convoluted data structure to depict the surface. In this model paradigm, type of surface have construction method that correspond to their geometric specification. Type of surface are, for instance, flat surface, developing surface, free form surface, revolution surface ect. It is important to notice that modelling paradigm is different to the modelling paradigm provided to the graphic card. This modelling paradigm is only used to offer a description system more suitable for deception. To display this depiction, all the model is translated to a mesh and texture model. Hardware of graphic cards can only handle this low level description format.

C.1.1.6 Constructive Solid Geometry

CSG C.6 is depicting a volume as combination of simple solid. Where Brep is a stitching combination of surfaces that are combination of lines themselves combination of points, CSG applies consecutively boolean operation to various solid in order to represent a volume. The Brep present a hierarchical combination but the stitching is spatially organised through the adjacencies of surface. In CSG the order and nature of combination matters. It follows that the data structure is drastically different. The data structure of CSG is a tree structure. The tree contain primitive elements and operator. With this simple grammar shape are constructed by sequential operation. The geometric data are 3D point of the primitive element center and its parameters. A separated group contain sequential operations between these geometric data. The advantages of CSG

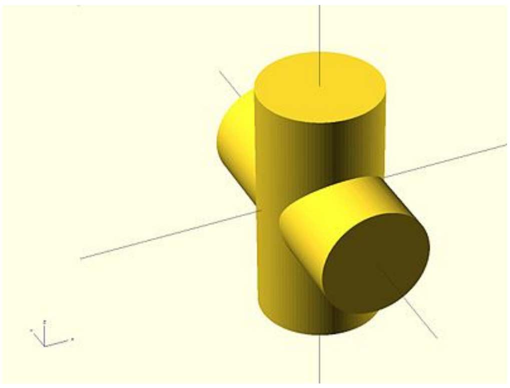


FIGURE C.6: Union of two tube section in a Solid Geometry modeller

representation is the impossibility to describe perfectly flat object without thickness. Such object can't exist naturally and can't be fabricated by any machine. By using such representation system instead of B-Rep, it ensures materialisation of the shape. [2]

[2] It doesn't ensure structural quality or fabrication

C.1.2 Layer for enrichment of model

The scale of computable representation paradigm is not linear. It becomes linear when a purpose is projected through the operations we want to perform. For instance, if the user cares about constructibility with a 3D printer, using a CSG representation paradigm to depict a object is better than using a mesh representation paradigm. In the other hand, if the user want to make a CG animation, mesh paradigm present less constrains for modelling. Above that mesh model can be enriched by a skeletal for animation. This skeletal is a enrichment of the model: it depict what part can move and how it moves regarding the other. This and other enrichment can be used to upgrade the 5 previous representation paradigm. For this reason, the author can say that CSG is better for design related to industrial fabrication but Collision Map or BIM enrichment feature are not essentially better than CSG representation. It will depends on the fabrication process and the type of data the CNC system can process correctly. Nonetheless, in this thesis, we propose a general scale that grade complexity of the data structure. This orientation is chosen because data structure is the matter on which computational designer work on and with. In the scale of representation paradigm, a threshold is passed when the representation paradigm is specialized for a domain knowledge. The appendix present here layer of information that can be added on a representation model paradigm to enrich it. They are organised by order of complexity. It's important to notice that since the data refinement is more complex due to a domain specialisation, it is always easier to downgrade from a semantic enriched format to geometric format. No information need to be added by an operator to produce translation of the 3D shape from one paradigm to a lower one. Sole the enrichment provided by the layer will be lost.

C.1.2.1 Collision Mesh

Collision MeshC.7 is a simple example to show what kind of enrichment of the model can be added. The Collision Mesh is mainly used in video games to define which part of the environment through which the player can travel through. It's usually a simpler copy of the original model. This model is on the same coordinate than the original one but on another "layer". Collision Mesh is often applied on surface model to indicate roughly the inside and the outside of an object. Usually a simple primitive shape will be used to make a bounding box of the model. Primitive are chosen because they ensure "watertight" volume and their collision is easy to compute. The downside is a slightly loss of precision. With a Collision Mesh a surface model that represent a volume gain almost the same advantages than a CSG model: The inside and the outside of the volume is always computable. But compute collision of CSG model is demanding in time computation, with actual hardware, it is impossible to compute collision in real-time.

The main advantages of Collision Mesh is its simplification of the mesh representation for collision purpose. So even on CSG model, a collision volume can be approximatively set around the object on a separate layer in order to simplify collision. For precise simulation purpose, approximation of collision map is not acceptable. In this case, the animation is not processed in real time. Note that collision is not taking deformation or resistance of object. They are processed as unbreakable solid objects.

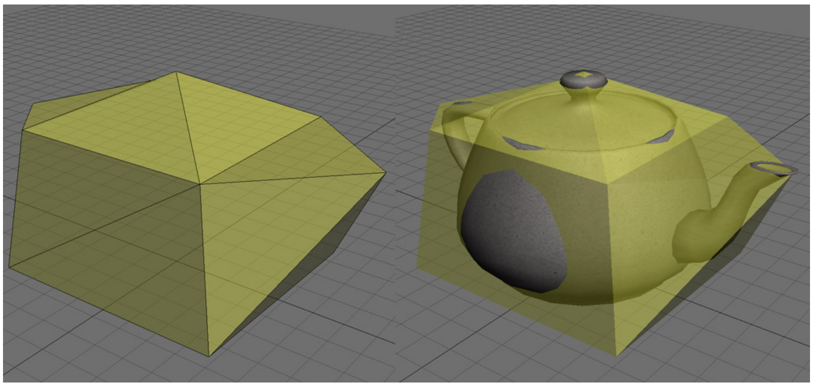


FIGURE C.7: In yellow we have the bare Collision Mesh of the more sophisticated model of the teapot.

C.1.2.2 BIM

The BIM is a centralisation of information about the modelled object. It gathers geometric informations, relational informations and semantic informations. The data structure of model becomes more complex and diversified. The BIM model use IFC format to standardise the description. Since it's exchange format, its standardisation is an important stake. Other paradigms of representation showed before are made for depiction and visual representation. They don't aim to be a standard, so different implementation of the paradigm are possible. For instance, depending on the final goal of the representation system, the implementation aims to improve the synergy between the model paradigm and the enrichment layer added to it. Note that the IFC is a object oriented file format. This is far more complex than format structure described before. Previous structures were close to the primitive data structure created in computer science. They are close to computer hardware working process and are easily implemented. It ensures efficient computation of these data.

C.1.3 File format

In practice of modelling software, the 3D format can mean the file format. Despite the file format internal organisation varies between a ".3ds"[3] and ".c4d"[4], this doesn't express a real difference in the paradigm that is used to establish the model. A file format correspond only to the software that produced the file. On a theoretical aspect, format are paradigm of geometry model —way to translate in a geometrical formalism. The file format are more about an implementation of such paradigm. It deals with hardware constraint for efficiency and reliability concern.

C.2 Generate 3D model from existing objects

Representation model can be produced *manually* by an operator. The reference of this representation can be a real object already existing or new object that is still in the mind of the operator. The purpose of modelling is to create a model that depict that imaginary object. Another way to create a 3D model is to directly capture it from the real world. With various techniques and devices it is possible to generate 3D model in different format. Capturing 3D model can help for modelling if the model to create is partially inspired from other real objects. Parts of capture of 3D model can be assembled and modified in order to create a 3D model. The following part will present different capture device and their raw output data format. Then capture principle section will explain how algorithm creates 3D model from these raw data.

C.3 Capture Device

This list of device doesn't aim to be exhaustive. It relates the various devices that the author had to use or to study during his research. They all provide data useful for automatic 3D modelling. The capture is more or less autonomous.

C.3.1 Photography

The most reliable and ancient device to capture 3D information. Originally capturing only 2D images, DSLR are 3D capturing device because with more than one picture of an object side, algorithm can extract depth information from them. DSLR are widely spread, relatively cheap and the quality of 2D raw image reach high resolution. These

[3]The file extension of 3D Studio Max Software
[4]The file extension of Cinema 4D

devices are not very autonomous since the operator of the capture need technical knowledge to take pictures that can be optimally processed by three-dimensional structure estimation algorithms. Even with perfect pictures, some type of object's structure can't be estimated. This is mainly due to the type of algorithms that can process 2D pictures. The autonomy can be increased by using techniques of HDR captures. It allows to takes all the details of a scene even in backlight conditions. It may help in some cases to soften shadows for pictures taken under a bright sun. All the requirement for taking good pictures for photogrammetry are explained in Digital photogrammetry [88]. They can be classified in two types of recommendations : shooting and colorimetry. To simplify the shooting, 360 camera can be used. And to automatise the triggering, video recording can be used.

C.3.2 Plenoptic Camera

Plenoptic Camera C.8 belongs to the category of computational photography:optical capture is merged with digital computation. To delineate digital computation involved in computational photography compare to the part of computation in digital photography, we can define that all computation that involve more than one image to produce new data can be considered as computational photography. For instance, stitching pictures to makes panorama or HDR tone mapping are considered as part of computational photography. In the case of Plenoptic Camera, the computation is used to create light field images. Like DSLR camera pictures, light field pictures contains color information of the ray of light that touch one APS of the CMOS. Add to this data, the plenoptic camera provide the field of the light. The field of the light field is a group of vector that indicate the flow direction of light. Despite light field are a rich of informations, they doesn't help much for 3D imaging. Nonetheless, it's possible to calculate depth map from them.

FIGURE C.8: Lytro Illum Camera is the first commercial light field Camera.

C.3.3 ScanNect, structured light

ScanNectC.9 and other structured light systems are range imaging technologies. This technology creates produce depth map from a specific point. Usually the optical center of the device. Those system are composed by one emitter and one sensor. The emitter project a pattern on the surface meant to be captured. The sensor capture back the pattern. This pattern is deformed by the relief of the surface. By comparing the original pattern and the deformed one the system's algorithm compute the range of each pixels of the image. The pattern is projected and captured through infra-red light in order to avoid perturbations inducted by the color of the captured surface. The resolution of the result depends on the complexity of the pattern and resolution of the infra-red camera sensor. Since the pattern is projected, the resolution of the pattern diminish when the surface is far. It follows that structured light system have a limited range that goes from a minimal distance to a maximal distance. In the ScanNect case, the sensor can capture between 50 cm and 3 m. Other restriction of the projection of structured light in infra-red is its inefficiency outdoor.

Figure C.9: This 3D scanning module is plug on the IPad and allow to scan continuously

C.3.4 Laser Scan

The most precise and efficient capture system. It throw a laser beams in all directions and calculate the time of flight to receive back the beam. In the same time, it takes picture in all direction in order to get the color of each point.This system produce directly coloured point cloud. The system need to stay static on a place during few minutes to capture all the space around. It's possible to complete the scan by moving the tripod on another place. To allow the system to track the previous position, marker should be place from the first scan session. Those markers stay in the same place when the tripod is moved in order to be used as reference point when different scan will be merged.

C.4 System based on 2D pictures capture principles

Devices such as Laser Scan, ScanNect or Lytro camera presented here offers directly 3D data in their output format. They are proprietary data format and code that produce the output is only accessible through API or proprietary software. None

C.4.1 Structure from Motion SfM

Structure from Motion is a technique based on geometric methods. The process to produce 3D model rely on image analysis and geometric projection techniques. The first step is detection of feature detection in the image. Then based on this feature position of the camera that took this picture is deduced during the bundle adjustment phase. Then when feature points and camera are correctly adjusted the dense cloud is created. Finally the points of the cloud are transformed in mesh with poisson or other algorithms.

C.4.1.1 feature detection

In a pictures humans notice specific point that can be easily identified. For instance, human brain is well trained to detect eyes. For computer vision there is similar feature that can be detected in a image. These feature are usually called by the name of the researcher who created the mathematical description of the feature. For instance, the Harris & Stephens or Sobel feature can be detected by specially tuned detector. This description state relation between gray values of one pixels and the surroundings ones. The first step is to transform the picture in grey scale because such systems are designed to work on level of grey pictures. Way to convert in level of grey can have impact on quantity of feature detected. The advantage of those type of feature detector are their reliance to rotation, and deformation. It means that on two different picture, the system detect the same feature. The features have an identity build on the value between central pixel and the surrounding pixels. This group of all the values make a vector that work like an identification of the feature. By this way, two pictures can be correlated if they contains features with the same identity. When the feature of all pictures are detected the Bundle adjustment can start.

C.4.1.2 Bundle adjustment

The Bundle adjustment algorithm places virtual camera in space according the pictures. The camera is at the same coordinate and orientation than it was when the picture has been taken. The process of this algorithm is to first place a camera in space and project the features points from the pictures toward infinite line. Then, a similar feature from another picture is taken and with the same process projected toward infinite. Since the feature are identical, the two infinite lines should cross somewhere in space. All similar feature from different pictures are correlated by this way. Gradually the position of the picture and virtual camera bundles are adjusted in the space in order to create coherence of the various intersections. The whole system is evolving until an equilibrium is reached. When the adjustment of bundle stop to move or move from a very short distance this equilibrium is reached.

C.4.1.3 Dense cloud

C.4.2 Simultaneous Localisation and Mapping SLAM

SLAM[89]

Appendix D

summary of Architecturology principles

Architecturology is a language to describe the design process in architecture. It has been set-up by Phillipe Boudon in XXX. This appendix doesn't expect to cover all the Architecturology system but will provide the key concept useful to understand the Thesis. The thesis uses terms of Architecturology because they are more precise to describe the design process. Those terms are general enough to describe any design process. It doesn't suppose that there is only one design process or that design should be done in certain way. Beside that, it is important to understand that Architecturology doesn't talk about creativity or what's going on the mind of the designer. It tries to give a factual descriptions of mental operations conducted during design process. Based on these definition discourse about design process can be constructed. For instance, in this thesis, the focus is on the relation with digital tool during algorithmic design process. It uses the Architecturology vocabulary to frame precisely when the digital tools have impact on the design process. As presented in the thesis, in most case, the digital tools helps to "give measurement". The creativity or the "hierarchy of scales" in the design process are note established with the help of digital tools yet. The vocabulary used will be clarified through the different section of this appendix.

D.1 Goal of Architecturology

Architecturology aims to be offer a scientific approach of the design. It repels the idea of design as a black box that can't and shouldn't be deciphered. To understand what happens in the mind, various approach co-exist and complete each other: psychology, neuroscience, sociology, biology, philosophy. They all offer valuable insight about where, what, how, when ideas comes. Architecturology takes a macro point of view. It describe mental space and moment that are not existing actually. They are theoretical construction that allow to describe the process. Within the moment and space described by that language, others science can set-up research and experiment to define where, what, how and when things happens. For instance, in the neuro-science field, the question is mainly studied. Nonetheless, if the model of the design process is not clearly established, creativity may be wrongly attribute to design operation on a specific scale. Architecturology is a close analysis of the design process that propose category to think about design. Beside it's scientific goal, it offers a pedagogical issue by providing a language to teach what can be verbally teach about the design process.

D.1.1 Epistemology concern

The original approach of Boudon when he invented architecturology was to propose a scientific field for architecture research. Indeed, we can wonder what is the science of architecture. More precisely, what is the science specific to architecture practice. Because if we look closely to all sciences involved in architecture, none of them are specific to architecture. They all relates to fundamental science that have paradigm and experimental process well defined. Architecture is originally a craft that had scientific knowledge but no academic knowledge production process. Fundamental science had paradigm and theories because they was in the seek to unveil something about nature. In architecture, what can be the closest to paradigm and theory are architectural movement and style. Theories in architecture are normative more than explicative. They go along with a political vision of the society and critical thinking about art trend. Architecture theories are not scientific in the sense that they are not testing a paradigm. If we look on the building science, construction and structural research are conducted by engineering department. They could be what architecture science practice should be but in the 18th century, the engineering section have been introduced.

D.1.1.1 Scientific crisis of the Architecture research

According to Thomas Kuhn in is seminal book, "the structure of scientific revolution"[90], science and pseudo-science can be differentiated through their historicity. A field of knowledge that doesn't evolve, that doesn't have an history is not a science. The typical structure of a scientific field is the implicit establishment of a paradigm between the researchers. Based on these paradigm, researchers produce theories to explain certain phenomenon. Then, with experiment those theories are tested. If new observations tends to contradict prediction that can be made with the theory. This theory is invalidated.

When observation that contradict the paradigm itself are accumulated, the paradigm starts to fall apart. The paradigm shift when a theory that goes against the paradigm can explain observations that are unexplained by other theories of the former paradigm. If we try to apply this structure to the architecture field in order to see if their is a historicity in the architecture science we can hardly find observation that goes against architecture theories. There is an history of architecture made of paradigm and theory but this is not an epistemology of architecture. Shift of paradigm in architecture are mainly caused by the evolution of the zeitgeist rather than objective observation that invalidate them. Usually, architectural theory disappear from the public debate when the architect or architecture critics stop to sustain it. Those theory are mainly practice motto rather than real issue "true" knowledge statement.
To establish architecture science, P. Boudon wonder what is the fundamental aspect of architecture. One aspect that can aspect that can be studied but can't be reduced to a sub-topic of a fundamental science. The entry point of his thinking is to wonder : *What is architecture ?*. Mere but powerful question that set a frame for scientific research in architecture. The development of his thought can be summarized by this citation of Karl Marx :

> A spider conducts operations that resemble those of a weaver, and a bee puts to shame many an architect in the construction of her cells. But what distinguishes the worst architect from the best of bees is this, that the architect raises his structure in imagination before he erects it in reality.

Architecture is about artificial shapes or schema that have been partially raised in imagination before being erected or realised in reality. It follows that the core question of architecture research is how to explain this phenomenon of imagination oriented in to realisation. It can be argue that design in general is about this kind of imagination, so this science is not specific to architecture. To answer to that point, the definition of architecture science specialise its definition and narrow his field to design case that involve space at the human scale. The point of this section is not to make an extensive critic of Architecturology. The author thinks that some part are subject to discussion but agree on the main idea : a vocabulary is needed to talk about architecture design process. The author take the party that architecture research doesn't have consistency as a fundamental science if it is not extended to a science of design. In the actual context, architecture science is a interdisciplinary science that produce knowledge at the intersection of different fundamental science. Since separation of science field is a convention to rationalise the joint effort of scientific in knowledge production, architecture as interdisciplinary science has a purpose to compensate that over-specification by recreating bounds between science.

D.2 Distinction between real space and conception space

To explain design process, Architecturology start to make a clear distinction between the real space and the conception space.[1]. Conception space is a mental space where the designer project his idea. In this space operation are processed without any impact and can be freely done and undone. No feedback are felt when ideas are interacting in this space. Only the critical thinking of the designer can extract meaningful insight form those operations processed in the conception space.

The real space is the reference space where things imagined in the conception space will be realised. It is a space that will provide feedback when the architect will realise the building he had in mind. This space as teleological goal frame the thinking in the conception space.

D.3 Giving measurement

Concretely, what the architect is doing in this conception space? According to Boudon's work he is attributing measure. Measure in measure attribution shouldn't be limited to distance measure. It should be understood in a larger sense of attributing a value according to a determined scale. Based on this definition of *measure attribution*, architect can attribute measure through the selection of pattern for a wall or the orientation of a building. To attribute a measure, the architect use the Dimension, Reference, Pertinence model.

D.3.1 Dimension, Reference, Pertinence Model

This model describe the process of measure attribution. To explain clearly this process the text will set a parallel between the theoretical description and a practical example. First the designer set a dimension within which he[2] is going make the measure attribution. This Dimension is the parameter on which architect is going to give measure. In the example D.1, the dimension is the height of the wall that will sustain stairs. To establish references D.2 that correspond to the dimension of height in meters, the architect will search in mind height of objects, people, furniture. These references can be the remains of his until now life experience or it can be references that he intentionally gather for this

[1]Conception has i in English a second meaning related to procreation. In this thesis when conception is used it's for underline a difference with "design" term that is more widely used in Anglo-Saxon architecture research field

[2]Despite, gender neutrality is the new norm for formal writing, the author will use "he" to refer to designer without presupposing that designer should be a male. Reader will by himself add "or she" to all occurrence of "he" pronoun.

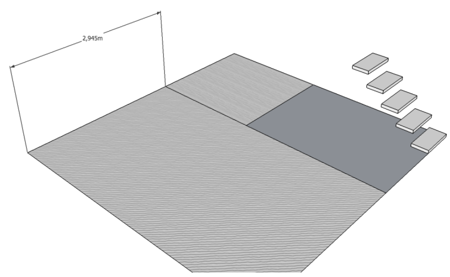


FIGURE D.1: Lets suppose that at this stage of the design process, the designer will attribute a measure. He chooses the height dimensions.

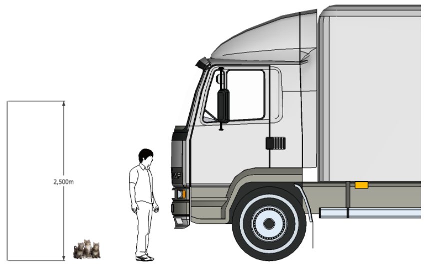


FIGURE D.2: To give a height measure, the architects gather the height of various references he already encounter.

measure attribution enterprise. Those references will serve as comparison points to decide the measure in the case he is attributing measure. Then, according to this reference and the context of the measure attribution he will chose a pertinent D.2 measure. A measure that suit to the situation according to his known references.

The action of giving measurement occurs during Conception Operation[91]. Indeed as said before, giving measurement can be done in various context, it is not only about deciding longitudinal size in cm of the object edges. It can be done on a symbolic level or for optical concern. Those categories are called *Scales* in Architecturology. In the next section we will see those Architecturologic scales and how the Conception operation is conducted within these scales.

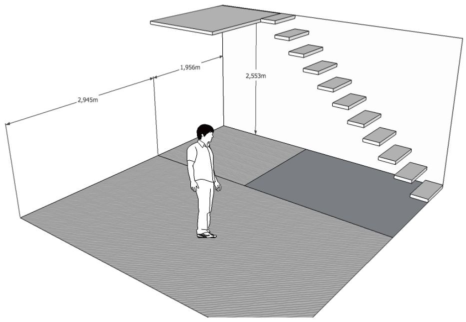


FIGURE D.3: Regarding the reference and the context of the real space, the architect select a measure in the scale of reference he appeals to define gradation in the selected dimension

D.4 Architecturologic Scales

There is 21 Architecturologic Scales and Y Meta-Scales. The system is open to the invention of new scales if it's necessary to describe a context of a conception operation.

- Economical scale
- Functional scale
- Semantic scale
- Optical scale
- Geometrical scale
- Visibility scale
- Symbolic shape scale
- Proximity scale
- Technical Scale
- Land plot Scale
- Model Scale
- Sociocultural Scale
- Geographic Scale
- Extension Scale

- Humane Scale
- Representation Scale
- Cartographic Scale
- Global Scale
- Symbolic dimensional Scale
- Level of design Scale
- Integration Scale

D.5 Conception operation

The conception operation is the attribution of a measurement in a architecturologic scale. This measure attribution happens in the conception space and is not necessarily keep in the final project. Conception operations are all step made by the designer to arrive to the final design. Those step are sequential in time. Even if the project go back to a later step, the design operations can be seen has a history of the design process. All this measure attribution are accumulated on a substrate model. The substrate is a theoretical model of the project. It's not necessarily the project physical model, the 3D model or the digital BIM model. Those model are representation of the substrate model that is in the mind of the designer. This substrate model is what receive the design operation. Each measure attribution are made on the substrate model in the conception space. By this naming convention, the Architecturology propose to describe design process.

D.6 Substrate model

To give an example of Architecturologic description of a design process, we could say : *In architecture project, the initial substrate mode can be the "model" of the site or a bare cube and it even can be a project brief. From this starting point the designer perform operation of conception and through representation, get a feedback on the choice made during the operation. Design is an alternation of conceptualisation and visualisation: the designer observe the representation, get an idea from this representation, apply it on his substrate model and create a visualisation.* The vocabulary is used to present a theory of architecture design : the alternation of visualisation an conceptualisation during the design and its subsequent representation phase.

Appendix E

Proposition of practical application of ConvNet in design tools : Semantic Rebuild Scan

E.0.1 EDGE project

During those 4 years, I encounter many opportunities. One of them was the EDGE workshop. I was from the beginning of my thesis interested by the actual production of something from my thesis. I was thinking about a software or a real scale model. EDGE program offer me this chance to be able to make a link between research and entrepreneurship. So drained from observations and thought in my research I came to the conclusion to design a product useful for my research and with potential for entrepreneurship. Technical and functional specifications of this "tool" are specified in chapter 4 and 5. Despite this tool couldn't be implemented because it requires expertise in computer science that goes beyond my academic speciality, it is precise and documented enough to assume it's proper working. At the time this thesis is written I'm looking with professionals in programming the implementation of the parts that couldn't be realised with my own skills.

E.0.1.1 Field Work

EDGE program is composed of a field work to meet professionals and specialist for advise on the entrepreneurship project. Thanks to EDGE project I succeed to present my project to Scientific and Entrepreneur. Professional such as 3D modelling unit in *Petit construction* or the CEO of *MiddleVR* validate my idea and ensure me of the existence of market for such product. On the fulfilment of the product, I met researcher of the GAMSEAU and IMAGINE for technical advice and feasibility enquiry.They had a positive opinion on feasibility but refrain me to program it alone. Through their technical advice, I progress on some technical part of the project but in the same time,I Figure that will be harder than expected and that it can't be done in the same time than a PhD. But since all the Research and Development are outcome of this research it's finally Integrated to the Thesis as an hypothesis for future development of the findings presented in this thesis.

E.1 Context of application

Before starting architectural design, designer need document of the site in order envision the important parameters. Local surroundings are often recorded by building surveyors. These records can be sometimes completed by visits to the site in order to get feeling of the place. Architects start to imagine possibilities for the future and need a model to test their idea. Records are abstractions of the real site that need to be reconstructed in the mind of the designer.

With a complex site, it can be hard to work out how to develop it. A good way to help the visualisation process can be 3D reconstruction based on a photography. Usually, architecture practice is about creating new building. It implies that existing old architecture around is demolished. This policy of Tabula Rasa is exactly the opposite to a continuous design that : " generate in the flow of what is already there [. . .]"[92]. This architecture practice can be considered, in the framework of the Algorithmic design, as minimal intervention in a system to solve a problem instead of fully create a costly solution from scratch. There needs to be a more comprehensive and integrated access to site data, in order to move toward this idealistic practice. This could be the key for an Algorithmic design sites intervention. For instance, instead of wasting energy to move away existing constructive elements, designer can build on them to carries and transfers weight. This is only possible if a accurate and editable model of the existing constructive elements is available document at the beginning of the design process.
Photogrammetry is a technology that is becoming more and more accessible. However, their internal structures are dissonant with our intuitive perception of spaces and objects, despite the fact that generated 3D models can be very accurate. In the end, it makes 3D shapes that are hardly editable for architects. For instance, if you want to change the radius of what you see as a tubular section, it is often necessary to remke the model with a built-in CAD software command upon the 3D scan. For a CAD user, such capture system create models in which, parameters of the shape can be modified

directly, without any remodelling or fuzzy vertices selection process. he can extrude a table or remove a window without carrying that they are in fact a group of vertices.

The 3D scan ease the precise and complete capture of geometric data. Unfortunately, it misses the semantic and constructive aspects. Previous research[93] has brought innovative solutions to cope with such problem. Without being too exhaustive: This work can be related to "Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques"[94] as much, Shen[95] and Andelo Martinovic[96] that present goals similar to this research. This short review presents the overall context and the challenges of 3D reconstruction. Here, we will explore a slightly different approach since it considers pictures as raw material and convolutional network as a technique to segment and identify the architectural component in the surroundings. Concerning the strategy, it may be sounds like "Structure Recovery by Part assembly"[95] except that it copes with the problem of detecting scattered parts of the component in various pictures instead of only one. Finally, these methods are slightly different because they use 3D point cloud analysis, computer vision oriented and shape grammar extraction but none of them use Deep Convolutional Neural Network (ConvNet)[97][98]. The experiments conducted in this paper will focus on architecture models of small volumes and enclosed spaces. Real scale can only be processed with large training data that is costly. So the training database is reduced to a number of easily testable cases. The results obtained using the limited database can serve to foreseen further scaling up to a larger system. Such extraction of information valuable for the design process can start from other type of raw data than picture. Starting from floor plan [99] or from depth map of urban area [65] can produce abstract data useful for nurturing the site problem analysis and with the right model[100], test hypothesis and explore design solution space.

E.2 Teaching the overlapping of stereo-vision

The objective to provide photogrammetry data set to a ConvNet is to teach a vision process close to the human one. Indeed, we don't see in three dimension. Technically this is two 2D pictures of the same scene from 2 point of view slightly different. This 2 point of view have always the same distance, the IPD. We will see that the system can avoid easily this constraint with the bundle adjustment. This compensate the interactive aspect of the vision that provide a lot of information about the position in depth of the object. Indeed, eye and head movement allow the human brain to instantaneously query the perception in order to get precision on the received information. Since the Data

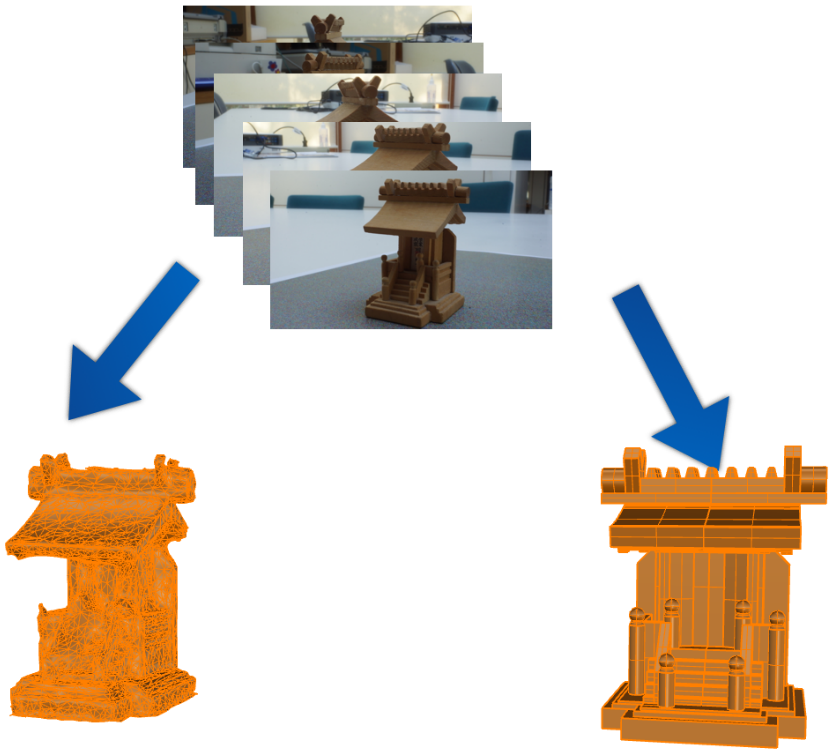


Figure E.1: With actual photogrammetry techniques, from a set of pictures it is possible to create a mesh like the one on the left side of the picture. With the use of ConvNet the SCS rebuild a model through advance modelling software command in a refined format such as B-Rep, CSG or even BIM

Set are static, the system can't benefit from such adjustment of the data-set. A third essential element to 3D space understanding of "completely new scene" is the short term memory. Indeed, when an object come to the field of view of a human for the first time, this one need to see the whole object from different angle. And thus, between those vision, the information of previous view of the object need to be stocked in order to be gradually cross referenced between them. By this way, the different view create a whole and various mental operation can be conducted with this volumetric idea of the object. For instance, evaluating if the object can pass through a door in width or height.

E.2.1 Teaching the object culture

Another aspect that help human to understand a scene is its long term memory. Indeed, to retrieve this volumetric idea of newly encounter object, a place is needed to stock them. Later, in a new scene, all objects already seen from one angle will be associated to the general volumetric idea of the object stocked in the long term memory. We will see later the structure and format of this memory work space for the scene understanding. This "memory" should be differentiated from the memory that store "past experience" of object. In fact, this memory is the ConvNet. More than one is necessary to achieve this goal of 3D reconstruction. The different ConvNets need to be articulated in order to perform the task. Above that, each ConvNet have a special architecture and training data set in order to perform the reconstruction task. The first ConvNet is called "Twin consensus segmentation ConvNet" (TwCS-ConvNet) and is trained to detect and link different faces of objects in a scene. The different object's face come from different point of view on the scene. As input it takes 2 pictures of the same scene with at least 50% overlapping images. The output is a segmentation of each pictures. The segmentation is done by object and semantic area such as sky or ground. The second ConvNet, is called "Topological Graph Parsing Recursive Neural Network" (TGP-RNN) is trained to create building sequences that creates 3D model of object that match a topological graph description. As input it takes a sequence of connected graph nodes and the output is a sequence of 3D software API commands. Note that second "ConvNet" is in fact a Recursive Neural Network (RNN). Other algorithms of CV and programs are used to prepare the input data of the first ConvNet and to bind together output of the first ConvNet and input of the RNN by transforming the format of the data.

E.2.2 General description of the Algorithm

The process to generate 3D model borrows some techniques of photogrammetry. It perform a similar task but by adding ConvNet between the intermediary steps of photogrammetry process, it creates new opportunities for designers. A normal photogrammetry software like subsection A.2.8 proceed in 4 steps:

- Feature extraction
- Camera Bundle Adjustment
- Dense Correlation
- Triangulation of the point cloud

The TwCS-ConvNet come after the Bundle Adjustment step and the TGP-RNN is created to replace the triangulation phase and come after the Dense Correlation phase. Computer Vision algorithms are used to analyse the segment of point cloud and transform them in graph that represent type of continuous surface and relation between surfaces. Then this graph is processed through the TGP-RNN that complete missing part of the graph by detecting elementary component of the volume that can be created with a command of preselected software. Finally complete graph are converted into BRep representation. If for instance, a CSG representation is desired, a different RNN should be used.

E.3 Preparation of the Input Data

The preparation of the input data for the system have the same requirement than photogrammetry one. Nonetheless, some preprocessing operation can improve and speed up the process. For instance, image that present close point of view can be gathered by cluster through a rapid color histogram comparison in the HSL model. For all pictures within a cluster feature are extracted and compared with the feature of all the images of the same cluster. Once the images are processed through the Feature extraction, the histogram cluster is use as can be abandoned and the picture classified by the rate of similar equivalent features. Indeed, features are not dependent of illumination or rotation and translation. Thus pictures that depict the same scene from close point of view may have a high rate of equivalent feature. The point of doing so is to avoid to proceed unrelated pictures in the TwCS-ConvNet in order to get relevant overlapping of pictures.

E.4 Camera coordinates extraction

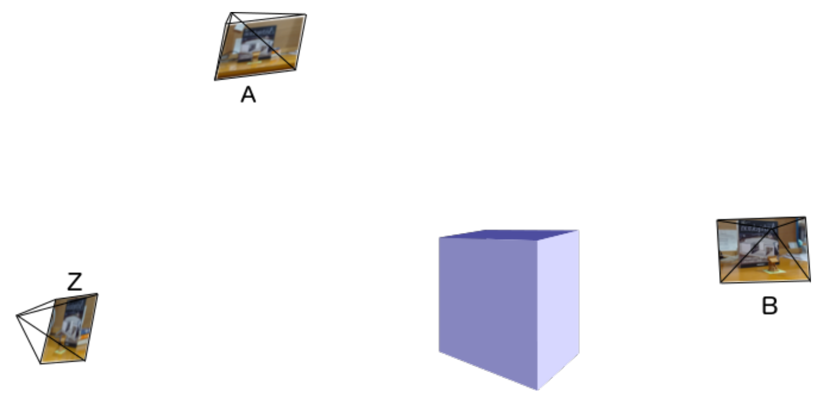


FIGURE E.2: This algorithm is used to localise the shooting coordinate of each pictures.

The point of the Bundle Adjustment step is to get the coordinates from where each pictures have been taken. Indeed, two pictures of the same scene embed implicitly their camera coordinate. These coordinate are valuable to help in the creation of depth-map. In a recent paper called, DeepStereo[101], the coordinates of Google car camera are use to deform pictures at different retro-projected position. Those retro-projection are simple perspective deformation of the pictures but by this way, they allow to transmit the camera position information compare to another image within the picture.
To understand why this technique require that images are deformed it is important to understand that ConvNet can process only uniform data format. The 3D coordinates of the camera position can't be provided in separate vector next to a picture images. Even if they can be all converted to unidimentional tensor matrix format, neural network

components are not will not be able to detec through the training the sinbularities of only one vector value amongst 3600 vector of 600 by 600 pixels RGB picture.The technique in John Flynn[101] paper is a way to implicitly embed camera coordinates and orientation within the pictures'. Like in Figure E.3, it applies a perspective deformation on a picture (B) in order to match with the point of view of a related picture (A). In other words, image (B) is re-projected with planar perspective deformation into the target of the camera's picture(A). In this case, the relative coordinates of the picture (B) according to the picture's (A) coordinates into the picture (B) are implicitly embed in the right picture of the Pair 1. Since the coordinates of the cameras' pictures are provided with internal and external calibration, pictures can be chained based on their geographic proximity through an hamiltonian graph of the triangulation of all Camera coordinates. Each picture will be re-projected in the subsequent picture space in the series according to the Hamiltonian path in the graph of the pictures.

E.4.1 Pairing pictures and Re-projection of the input image by pair

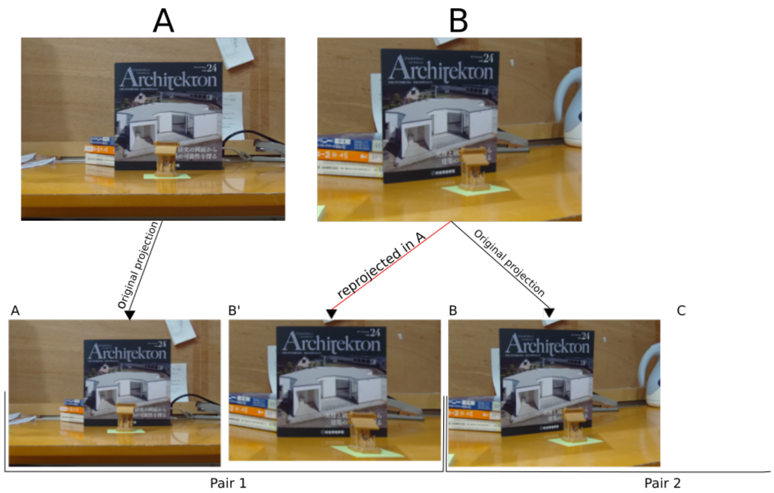


FIGURE E.3: Anamorphosis of the picture A from the point of view of the picture B

Indeed, the perspective deformation of a picture inform through the matrix deformation of the image the camera position relatively to another one. Such information helps the ConvNet to deduce rules concerning the similarity or absence of overlapping area. Indeed, this method that we can call "geometric" come to complete a "semantic" one that detect the category of a pixel like in the work of Farabert[87]. ConvNet for segmentation are usually supported by segmentation algorithm coming from the Computer Vision field. Such system are based on analyse of pixel value variation in an area in order to detect edges. This is a low level method and compare to the pixel category identification we could say that it is a purely mathematical method. It doesn't rely on semantic information learn by training but are based on hand crafted detectors. With such segmentation program, super pixels can be created. A super pixel is a mound of pixels that have similar values. The super pixel get is segment category by the vote of each identified pixels it contains. If two images are segmented with standard segmentation ConvNet, the segment of one picture will not be correlated. They will be independent in each picture and thus it will be impossible to say which segment of the same category belong indeed to the actual same object. Embedding camera information allow to detect overlapping part and thus to chain the segment together from picture to picture. In the end each object are recorded as a chain of segment as the Figure E.4.

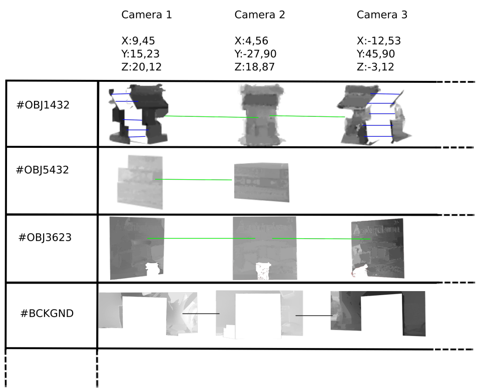


FIGURE E.4: This Table show symbolically how each object are represented in the memory of the system

E.5 Architecture of the ConvNet : Twin Segmentation Dataset

The TwS-CNN is illustrated in the Figure E.5. It reuse classical layer existing in [87] such as Spatial Pyramid that extract information on the general layout of the picture and traditional Convolution Map to create Feature detector related to the category to detect. The twin input entry merge their information in a final Fully Connected (FC) layer. To set-up this network, the upper layers will be train on simple segmentation then the network is copied and the last FC is removed. Then, the TwS-CNN with its double input is trained with the data set specific for twin segmentation.

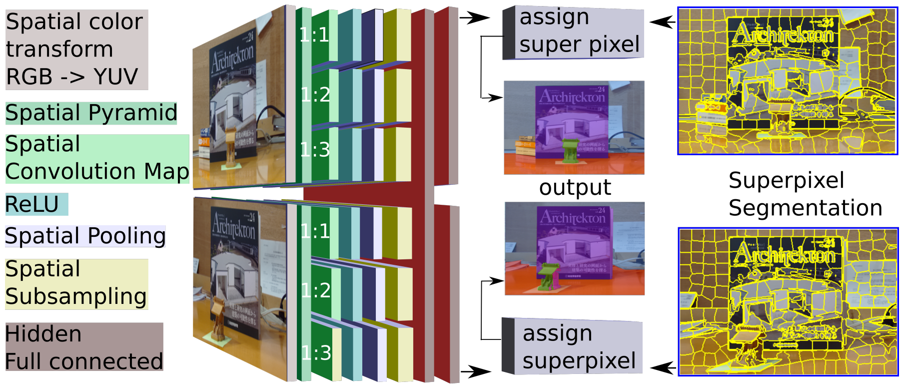


FIGURE E.5: Twin Segmentation ConvNet

E.6 Training Data set

For this tow step training, 2 data set will be needed, one to train to segment and one to train to detect overlapping and segment them. The first data set can be found easily since segmentation is a task now very common for ConvNet research comunity [102]. The second one need to be created through by taking pictures and creating ground truth segmentation for the learning. To ease the task, existing photogrammetry data set can be use. Then, using existing segmentation ConvNet can be use to automatically pre-segment pictures. Then manually with and Computer vision segmentation algorithm such as SLIC, QuickShift or Felzenszwalb (see section A.2.9 and more interactive one such as GraphCut. Then for optimisation of the task completion, the segment can be propagated amongst pictures representing similar things. A a plug-in of MicMac [86] recently developed is used to perform this task.

E.6.1 Acquisition of the Data set

Different capture device have been tried to directly get picture and depth map in order to save some work for the future user. Plenoptic camera and projected light capture system have been used but didn't show reliable performance. Finally, it was more efficient to use photogrammetry to create the depth map by creating a mesh then extracting the depth map form points of view identical to camera that took the original photo.

E.7 Parsing of the Depth-map strip

Then to construct this shape, there are various recipes. The point is to make them match with avaible measurement in the vectorial database. In the ends, it proposes various arrangements sorted by a range of probability. The user selects the assembly process the most interesting for the use he wants to make. To store this possibility of reconstruction, an adjacency graph is generated from the scanned letters. From this adjacency graph various proprieties of the shapes can be detected.:Parallel surfaces, revolution shape, convexity.

References

[1] Horst W. J. Rittel and Melvin M. Webber. Dilemmas in a general theory of planning. *Policy Sciences*, 4(2):155–169, June 1973. ISSN 0032-2687, 1573-0891. doi: 10.1007/BF01405730. URL http://link.springer.com/article/10.1007/BF01405730.

[2] Jay Doblin. A short, grandiose theory of design. *STA Design Journal, Analysis and Intuition*, pages 6–16, 1987. URL http://www.chicagodesignarchive.org/pdfs/sta_design_journal_jaydoblin.pdf.

[3] Nicholas Negroponte. *Being Digital*. Knopf Doubleday Publishing Group, January 2015. ISBN 978-1-101-91182-2.

[4] Mark Rowlands. *The New Science of the Mind: From Extended Mind to Embodied Phenomenology*. MIT Press, August 2010. ISBN 978-0-262-28894-1.

[5] Lambros Malafouris. *How Things Shape the Mind*. MIT Press, July 2013. ISBN 978-0-262-01919-4.

[6] Arnold H. Buss and Robert Plomin. *A temperament theory of personality development*. Wiley-Interscience, 1975. URL http://psycnet.apa.org/psycinfo/1975-29681-000.

[7] Vilém Flusser. *Gesten: Versuch einer Phönomenologie*. FISCHER Taschenbuch, Frankfurt am Main, 1994. ISBN 978-3-596-12241-7.

[8] Margaret A. Boden. *Artificial Intelligence and Natural Man*. Basic Books, 1977. ISBN 978-0-465-00450-8.

[9] Jeannette M. Wing. Computational thinking. *Communications of the ACM*, 49 (3):33, March 2006. ISSN 00010782. doi: 10.1145/1118178.1118215. URL http://portal.acm.org/citation.cfm?doid=1118178.1118215.

[10] Joi Ito. Design and Science. *PubPub*, January 2016. URL http://jods.mitpress.mit.edu/pub/designandscience.

[11] Christopher Alexander, Sara Ishikawa, and Murray Silverstein. *A Pattern Language: Towns, Buildings, Construction*. OUP USA, 1977. ISBN 978-0-19-501919-3.

[12] Christopher Alexander. *The Timeless Way of Building*. Oxford University Press, New York, 1979. ISBN 978-0-19-502402-9.

[13] Donald A. Schön. *The reflective practitioner: how professionals think in action*. Arena, September 1995. ISBN 978-1-85742-319-8.

[14] Margaret A. Boden. *The Creative Mind: Myths and Mechanisms*. Psychology Press, 2004. ISBN 978-0-415-31452-7.

[15] René Descartes. *Discourse on method and meditations on first philosophy*. Hackett Publishing, 1999. URL https://books.google.com/books?hl=fr&lr=&id=o1buCgAAQBAJ&oi=fnd&pg=PR3&dq=discourse+of+method+descartes&ots=C05UEeqJXx&sig=BI32AVBuS-f2jr52UqiGXJ21Lxg.

[16] Kostas Terzidis. *Algorithmic Architecture*. Routledge, Amsterdam ; Boston, 2006. ISBN 978-0-7506-6725-8.

[17] Herbert A. Simon. *The sciences of the artificial*. MIT press, 1996. URL https://books.google.com/books?hl=fr&lr=&id=k5Sr0nFw7psC&oi=fnd&pg=PR9&dq=science+of+the+artificial+Simon&ots=-v1KpIDNKu&sig=OnsoAl02XGwh5hSFgMSpvml8New.

[18] Philippe Boudon. *Sur l'espace architectural: essai d'épistémologie de l'architecture*. Parenthèses, Marseille, 2003. ISBN 2-86364-621-4 978-2-86364-621-2.

[19] Geraint A Wiggins. Searching for computational creativity. *New Generation Computing*, 24(3):209–222, 2006.

[20] Alison Pease, Daniel Winterstein, and Simon Colton. Evaluating machine creativity. In *Workshop on creative systems, 4th international conference on case based reasoning*, pages 129–137, 2001.

[21] Selmer Bringsjord, Paul Bello, and David Ferrucci. Creativity, the Turing test, and the (better) Lovelace test. pages 215–239, 2003. URL http://link.springer.com/chapter/10.1007/978-94-010-0105-2_12.

[22] Nancy J Nersessian. *Creating scientific concepts*. MIT press, 2010.

[23] J. Christopher Jones. *Design methods*. Van Nostrand Reinhold, New York, 1992. ISBN 0-442-01182-2 978-0-442-01182-6 0-471-28496-3 978-0-471-28496-3. OCLC: 25410842.

[24] James J. Gibson. *The Ecological Approach to Visual Perception: Classic Edition*. Psychology Press, November 2014. ISBN 978-1-317-57937-3.

[25] Christian Gänshirt. *Tools for Ideas: An Introduction to Architectural Design*. Birkhäuser Architecture, Basel ; Boston : London, 1 edition edition, June 2007. ISBN 978-3-7643-7577-5.

[26] Nicholas Davis, Chih-Pin Hsiao, Yanna Popova, and Brian Magerko. An Enactive Model of Creativity for Computational Collaboration and Co-creation. In *Creativity in the Digital Age*, pages 109–133. Springer, 2015.

[27] Ben Shneiderman, Gerhard Fischer, Mary Czerwinski, Mitch Resnick, Brad Myers, Linda Candy, Ernest Edmonds, Mike Eisenberg, Elisa Giaccardi, Tom Hewett, and others. Creativity support tools: Report from a US National Science Foundation sponsored workshop. *International Journal of Human-Computer Interaction*, 20 (2):61–77, 2006.

[28] Aurélie De Boissieu. *Modélisation paramétrique en conception architecturale : Caractérisation des opérations cognitives de conception pour une pédagogie*. PhD thesis, Université Paris Est, 2013.

[29] Erin Cherry and Celine Latulipe. Quantifying the Creativity Support of Digital Tools Through the Creativity Support Index. *ACM Trans. Comput.-Hum. Interact.*, 21(4):21:1–21:25, June 2014. ISSN 1073-0516. doi: 10.1145/2617588. URL http://doi.acm.org/10.1145/2617588.

[30] Kumiyo Nakakoji. Meanings of tools, support, and uses for creative design processes. In *International Design Research Symposium '06, Seoul, Korea*, 2006.

[31] Todd Lubart. How can computers be partners in the creative process: classification and commentary on the special issue. *International Journal of Human-Computer Studies*, 63(4):365–369, 2005.

[32] Rivka Oxman. The thinking eye: visual re-cognition in design emergence. *Design Studies*, 23(2):135–164, March 2002. ISSN 0142-694X. doi: 10.1016/S0142-694X(01)00026-6. URL http://www.sciencedirect.com/science/article/pii/S0142694X01000266.

[33] Gabriela Goldschmidt. The dialectics of sketching. *Creativity Research Journal*, 4 (2):123–143, 1991. doi: 10.1080/10400419109534381. URL http://dx.doi.org/10.1080/10400419109534381.

[34] Donald A. Schon and Glenn Wiggins. Kinds of seeing and their functions in designing. *Design Studies*, 13(2):135–156, April 1992. ISSN 0142-694X. doi: 10.

1016/0142-694X(92)90268-F. URL http://www.sciencedirect.com/science/article/pii/0142694X9290268F.

[35] Masaki Suwa. Constructive perception: Coordinating perception and conception toward acts of problem-finding in a creative experience1. *Japanese Psychological Research*, 45(4):221–234, November 2003. ISSN 1468-5884. doi: 10.1111/1468-5884.00227. URL http://onlinelibrary.wiley.com/doi/10.1111/1468-5884.00227/abstract.

[36] Benay Gürsoy and Mine Özkar. Visualizing making: Shapes, materials, and actions. *Design Studies*, 41, Part A:29–50, November 2015. ISSN 0142-694X. doi: 10.1016/j.destud.2015.08.007. URL http://www.sciencedirect.com/science/article/pii/S0142694X15000617.

[37] Robert Woodbury. *Elements of Parametric Design*. Taylor and Francis, 2010. URL http://eprints.iat.sfu.ca/930/.

[38] Amy Briggs Owen Astrachan. The cs principles project. *ACM Inroads*, 3(2), 2012. ISSN 2153-2184.

[39] Bryan Lawson. *How designers think: The design process demystified*. Routledge, 2006. URL https://books.google.com/books?hl=fr&lr=&id=7pNxxZRCLmcC&oi=fnd&pg=PR1&dq=+How+designers+think&ots=VteiZXT_gN&sig=cVt0dh5K8Fb6rgXdK5se96N_DkI.

[40] Jesko Fezer, Martin Schmitz, and Lucius Burckhardt. *Wer plant die Planung?: Architektur, Politik und Mensch*. Schmitz, Martin, Berlin, January 2004. ISBN 978-3-927795-39-6.

[41] Steve Easterbrook. From Computational Thinking to Systems Thinking. *Proc. ICT4S*, 2014. URL http://www.atlantis-press.com/php/download_paper.php?id=13446.

[42] Shoei Matsukawa and 000lab. arkitektome. http://arkhitektome.com, 04 2013. URL http://arkhitektome.com.

[43] A. M. Turing. Computing Machinery and Intelligence. *Mind*, 59(236):433–460, 1950. ISSN 0026-4423. URL http://www.jstor.org/stable/2251299.

[44] Etienne Benson. Intelligent intelligence testing. *Monitor on Psychology*, 34(2):48–58, 2003. URL http://otsr.virtualave.net/Hammond/Intelligent_Intelligence_Testing.doc.

[45] James R. Flynn. The mean IQ of Americans: Massive gains 1932 to 1978. *Psychological Bulletin*, 95(1):29–51, 1984. ISSN 1939-1455(Electronic);0033-2909(Print). doi: 10.1037/0033-2909.95.1.29.

[46] Mark O. Riedl. The Lovelace 2.0 Test of Artificial Creativity and Intelligence. *arXiv:1410.6142 [cs]*, October 2014. URL http://arxiv.org/abs/1410.6142. arXiv: 1410.6142.

[47] Huazheng Wang, Fei Tian, Bin Gao, Jiang Bian, and Tie-Yan Liu. Solving Verbal Comprehension Questions in IQ Test by Knowledge-Powered Word Embedding. *arXiv:1505.07909 [cs]*, May 2015. URL http://arxiv.org/abs/1505.07909. arXiv: 1505.07909.

[48] John R. Searle. Is the brain's mind a computer program. *Scientific American*, 262(1):26–31, 1990. URL http://sils.shoin.ac.jp/~gunji/AI/CR/Is_The_Brains_Mind_A_Computer_Program.pdf.

[49] Sandeep Rajani. Artificial intelligence–man or machine. *International Journal of Information Technology*, 4(1):173–176, 2011.

[50] Gilbert Simondon. *Du mode d'existence des objets techniques*. Aubier-Montaigne, 1969.

[51] Irving John Good. Speculations concerning the first ultraintelligent machine. *Advances in computers*, 6:31–88, 1966. URL http://www.sciencedirect.com/science/article/pii/S0065245808604180.

[52] A. Newell and H. Simon. The logic theory machine–A complex information processing system. *IRE Transactions on Information Theory*, 2(3):61–79, September 1956. ISSN 0096-1000. doi: 10.1109/TIT.1956.1056797.

[53] Allen Newell and Herbert A. Simon. Gps, a program that simulates human thought. In George F. Luger, editor, *Computation &Amp; Intelligence*, chapter GPS, a Program That Simulates Human Thought, pages 415–428. American Association for Artificial Intelligence, Menlo Park, CA, USA, 1995. ISBN 0-262-62101-0. URL http://dl.acm.org/citation.cfm?id=216000.216024.

[54] Horst Rittel and Douglas Noble. Issue-based information systems for design. *Univ. Calif. Berkeley Work. Pap*, 492, 1989.

[55] Alfred Whitehead and Bertrand Russell. *Principia mathematica*. The University Press, Cambridge, 2ème éd. edition, 1968.

[56] Claude E. Shannon. XXII. Programming a computer for playing chess. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 41 (314):256–275, March 1950. ISSN 1941-5982. doi: 10.1080/14786445008521796. URL http://dx.doi.org/10.1080/14786445008521796.

[57] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, January 2016. ISSN 0028-0836. doi: 10.1038/nature16961. URL http://www.nature.com/nature/journal/v529/n7587/full/nature16961.html.

[58] David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A. Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John Prager, and others. Building Watson: An overview of the DeepQA project. *AI magazine*, 31(3):59–79, 2010. URL http://www.aaai.org/ojs/index.php/aimagazine/article/viewArticle/2303.

[59] Donald Michie. Programmer's gambit. *New Scientist*, 17:329, 1972. URL https://saltworks.stanford.edu/assets/fk070vw3202.pdf.

[60] Kostas Terzidis. *Permutation Design: Buildings, Texts, and Contexts*. Routledge, 2014. URL https://books.google.com/books?hl=fr&lr=&id=yHhsBAAAQBAJ&oi=fnd&pg=PP1&dq=Kostas+Terzidis+permutation&ots=4P_OLbezR5&sig=-Z06wr0ENjAemwAUN82wu8Wzn_8.

[61] Alexander Mordvintsev, Christopher Olah, and Mike Tyka. Inceptionism: Going Deeper into Neural Networks, 2015. URL http://googleresearch.blogspot.com/2015/06/inceptionism-going-deeper-into-neural.html.

[62] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015. URL http://arxiv.org/abs/1508.06576.

[63] Karl DD Willis and Jacob Hina. Alchemy: experiments in interactive drawing, creativity, & serendipity. In *Proceedings of the seventh ACM conference on Creativity and cognition*, pages 441–442. ACM, 2009.

[64] Davis Daniel. *Modelled on Software Engineering: Flexible Parametric Models in the Practice of Architecture*. PhD thesis, RMIT, February 2013. URL https://www.researchgate.net/publication/301354977_Modelled_

on_Software_Engineering_Flexible_Parametric_Models_in_the_Practice_of_Architecture.

[65] Sonit Bafna. SPACE SYNTAX A brief introduction to its logic and analytical techniques. *Environment and Behavior*, 35(1):17–29, 2003. URL http://eab.sagepub.com/content/35/1/17.short.

[66] C. M. Herr, N. Gu, S. Roudavski, and M. A. Schnabel. Packing the "chinese box". In *CAADRIA 2011*, 2011. URL http://cumincad.architexturez.net/system/files/pdf/caadria2011_068.content.pdf.

[67] John Wood. The tetrahedron can encourage designers to formalize more responsible strategies. *Art, Design & Communication in Higher Education*, 3(3):175–192, December 2004. doi: 10.1386/adch.3.3.175/1.

[68] Nelson Cowan. The magical mystery four how is working memory capacity limited, and why? *Current directions in psychological science*, 19(1):51–57, 2010.

[69] Jacob Warren Getzels and Mihály Csíkszentmihályi. *The Creative Vision: A Longitudinal Study of Problem Finding in Art*. Wiley, 1976. ISBN 978-0-471-01486-7.

[70] Jeanne Nakamura and Mihaly Csikszentmihalyi. Flow theory and research. *Handbook of positive psychology*, pages 195–206, 2009. URL https://books.google.com/books?hl=fr&lr=&id=6IyqCNBD6oIC&oi=fnd&pg=PA195&dq=Flow+Theory+and+Research&ots=ILIbML3hvz&sig=lr-zEk-Q19HQWFIfSbnDKieBTpM.

[71] Örjan de Manzano, Töres Theorell, László Harmat, and Fredrik Ullén. The psychophysiology of flow during piano playing. *Emotion*, 10(3):301, 2010. URL http://psycnet.apa.org/journals/emo/10/3/301/.

[72] Giovanni B. Moneta. On the measurement and conceptualization of flow. In *Advances in flow research*, pages 23–50. Springer, 2012. URL http://link.springer.com/10.1007/978-1-4614-2359-1_2.

[73] Sherry Turkle. *Simulation and its discontents*. The MIT Press, 2009. ISBN 978-0-262-01270-6.

[74] I. Scott MacKenzie and Colin Ware. Lag as a determinant of human performance in interactive systems. In *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*, pages 488–493. ACM, 1993. URL http://dl.acm.org/citation.cfm?id=169431.

[75] Justin Johnson. neural-style. https://github.com/jcjohnson/neural-style, 2015.

[76] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv:1511.06434 [cs]*, November 2015. URL http://arxiv.org/abs/1511.06434. arXiv: 1511.06434.

[77] Ben Jonson. Design ideation: the conceptual sketch in the digital age. *Design Studies*, 26(6):613–624, November 2005. ISSN 0142-694X. doi: 10.1016/j.destud.2005.03.001. URL http://www.sciencedirect.com/science/article/pii/S0142694X05000189.

[78] Matthias Bethge, Alexander S. Ecker, Leon A. Gatys, Łukasz Kidziński, and Michał Warchoł. Visual turing test. Test: http://turing.deepart.io , Result: http://turing.deepart.io:3838, 2016.

[79] Deborah Chambers and Daniel Reisberg. What an image depicts depends on what an image means. *Cognitive Psychology*, 24(2):145–174, April 1992. ISSN 0010-0285. doi: 10.1016/0010-0285(92)90006-N. URL http://www.sciencedirect.com/science/article/pii/001002859290006N.

[80] Panagiotis Louridas. Design as bricolage: anthropology meets design thinking. *Design Studies*, 20(6):517–535, November 1999. ISSN 0142-694X. doi: 10.1016/S0142-694X(98)00044-1. URL http://www.sciencedirect.com/science/article/pii/S0142694X98000441.

[81] Paul M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of experimental psychology*, 47(6):381, 1954. URL http://psycnet.apa.org/journals/xge/47/6/381/.

[82] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR*, abs/1409.1556, 2014.

[83] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning Deep Features for Scene Recognition using Places Database. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 487–495. Curran Associates, Inc., 2014. URL http://papers.nips.cc/paper/5349-learning-deep-features-for-scene-recognition-using-places-database.pdf.

[84] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[85] Garry Kasparov. The chess master and the computer. *The New York Review of Books*, 57(2):16–19, 2010. URL http://www.aiecon.org/herbertsimon/series19/Kasparov_NYRB.doc.

[86] Adeline Manuel, Livio De Luca, and Philippe Véron. A Hybrid Approach for the Semantic Annotation of Spatially Oriented Images. *International Journal of Heritage in the Digital Era*, 3(2):305–320, June 2014. ISSN 2047-4970, 2047-4989. doi: 10.1260/2047-4970.3.2.305. URL http://hde.sagepub.com/content/3/2/305.

[87] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning Hierarchical Features for Scene Labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1915–1929, August 2013. ISSN 0162-8828. doi: 10.1109/TPAMI.2012.231.

[88] Yves Egels and Michel Kasser. *Digital Photogrammetry*. CRC Press, September 2003. ISBN 978-0-203-30595-9.

[89] Renato F. Salas-Moreno, Richard A. Newcombe, Hauke Strasdat, Paul HJ Kelly, and Andrew J. Davison. Slam++: Simultaneous localisation and mapping at the level of objects. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1352–1359. IEEE, 2013. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6619022.

[90] Thomas Kuhn. *The structure of scientific revolutions,*. University of Chicago Press, Chicago, [2d ed., enl. edition, 1970. ISBN 978-0-226-45803-8.

[91] S. Ben Rajeb, Caroline Lecourtois, and François Guéna. Operations of Conception in Architectural Collaborative Design. In *Proceedings of the eCAADe Conference*, pages 687–695, 2010. URL http://cumincad.architexturez.net/system/files/pdf/ecaade2010_124.content.pdf.

[92] John Wood. *Design for Micro-utopias: Making the Unthinkable Possible*. Gower, 2007. ISBN 978-0-7546-4608-2.

[93] Langyue Wang and Gunho Sohn. An integrated framework for reconstructing full 3d building models. In *Advances in 3D Geo-Information Sciences*, pages 261–274. Springer, 2011. URL http://link.springer.com/chapter/10.1007/978-3-642-12670-3_16.

[94] Pingbo Tang, Daniel Huber, Burcu Akinci, Robert Lipman, and Alan Lytle. Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques. *Automation in Construction*, 19(7):

829–843, November 2010. ISSN 0926-5805. doi: 10.1016/j.autcon.2010.06.007. URL http://www.sciencedirect.com/science/article/pii/S0926580510000907.

[95] Chao-Hui Shen, Hongbo Fu, Kang Chen, and Shi-Min Hu. Structure recovery by part assembly. *ACM Trans. Graph.*, 31(6):180, 2012. URL http://dblp.uni-trier.de/db/journals/tog/tog31.html#ShenFCH12.

[96] Andelo Martinovic. *Inverse Procedural Modeling of Buildings*. PhD thesis, KU LEUVEN, September 2015. URL https://lirias.kuleuven.be/handle/123456789/502580.

[97] Michael A. Nielsen. *Neural Networks and Deep Learning*. Determination Press, 2015. URL http://neuralnetworksanddeeplearning.com.

[98] Y LeCun, B Boser, JS Denker, D Henderson, RE Howard, W Hubbard, and LD Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems 2, NIPS 1989*, pages 396–404. Morgan Kaufmann Publishers, 1990.

[99] Emily J. Whiting. *Geometric, topological & semantic analysis of multi-building floor plan data*. Thesis, Massachusetts Institute of Technology, 2006. URL http://dspace.mit.edu/handle/1721.1/35123.

[100] Joaquim Silvestre. Game design for architecture design. In *ALGODE*, 2001.

[101] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. DeepStereo: Learning to Predict New Views from the World's Imagery. *arXiv:1506.06825 [cs]*, June 2015. URL http://arxiv.org/abs/1506.06825. arXiv: 1506.06825.

[102] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.