Application of Machine Learning within Visual Content Production

Daniele Giunchi

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

of

University College London.

Department of Computer Science

University College London

June 25, 2021

I, Daniele Giunchi, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Abstract

We are living in an era where digital content is being produced at a dazzling pace. The heterogeneity of contents and contexts is so varied that a numerous amount of applications have been created to respond to people and market demands. The visual content production pipeline is the generalisation of the process that allows a content editor to create and evaluate their product, such as a video, an image, a 3D model, etc. Such data is then displayed on one or more devices such as TVs, PC monitors, virtual reality head-mounted displays, tablets, mobiles, or even smartwatches. Content creation can be simple as clicking a button to film a video and then share it into a social network, or complex as managing a dense user interface full of parameters by using keyboard and mouse to generate a realistic 3D model for a VR game. In this second example, such sophistication results in a steep learning curve for beginner-level users. In contrast, expert users regularly need to refine their skills via expensive lessons, time-consuming tutorials, or experience. Thus, user interaction plays an essential role in the diffusion of content creation software, primarily when it is targeted to untrained people. In particular, with the fast spread of virtual reality devices into the consumer market, new opportunities for designing reliable and intuitive interfaces have been created. Such new interactions need to take a step beyond the point and click interaction typical of the 2D desktop environment. The interactions need to be smart, intuitive and reliable, to interpret 3D gestures and therefore, more accurate algorithms are needed to recognise patterns. In recent years, machine learning and in particular deep learning have achieved outstanding results in many branches of computer science, such as computer graphics and human-computer interface, outperforming algorithms that were considered state of the art, however, there are only fleeting efforts to translate this into virtual reality.

In this thesis, we seek to apply and take advantage of deep learning models to two different content production pipeline areas embracing the following subjects of interest: advanced methods for user interaction and visual quality assessment. First, we focus on 3D sketching to retrieve models from an extensive database of complex geometries and textures, while the user is immersed in a virtual environment. We explore both 2D and 3D strokes as tools for model retrieval in VR. Therefore, we

implement a novel system for improving accuracy in searching for a 3D model. We contribute an efficient method to describe models through 3D sketch via an iterative descriptor generation, focusing both on accuracy and user experience. To evaluate it, we design a user study to compare different interactions for sketch generation. Second, we explore the combination of sketch input and vocal description to correct and fine-tune the search for 3D models in a database containing fine-grained variation. We analyse sketch and speech queries, identifying a way to incorporate both of them into our system's interaction loop. Third, in the context of the visual content production pipeline, we present a detailed study of visual metrics. We propose a novel method for detecting rendering-based artefacts in images. It exploits analogous deep learning algorithms used when extracting features from sketches.

Impact Statement

The work described in this thesis contributed to the publications within DISTRO (Distributed 3D Object Design) Marie Skłodowska-Curie Innovative Training Network (ITN) that was funded by the European Union through Horizon 2020 research program.

Sketching in VR is a relatively new field and largely dedicated to creative activities, such as is the case for drawing in Google TiltBrush[1]. The ability to add incorporate external elements such as 3D models can be handled but requires moving from the current activity (sketching) to floating menus or gestures. In contrast, in this thesis in Chapter 3, we explore different interactive methods of sketching to search for a 3D model without interrupting the VR experience and also sketching activity. Such techniques could be added as plugin to either creative applications or professional modelling in the engineering and architecture disciplines. This impacts the professional activities by streamlining tasks without the need for context switching. In Chapter 4 the 3D immersive sketching is extended with the addition of an intuitive modality, speech, which provides different expressive power to improve accuracy and user experience. Our study highlights the possibility to use a coherent strategy for integrating further input modalities and can be additionally integrated into such content creation and professional tools.

Visual content production pipeline benefit from our work in Chapter 5 detailing an innovative model to assess image quality and identifying potential visualisation issues. This approach can be used within game engines to evaluate if downsampled textures can be used instead of the full-resolution version. Loading lighter assets helps the performance of products that run on devices with limited memory resources such as VR HMD. If such resources need to be fetched from the network, we reduce the bandwidth consumption and speed up the downloading process.

[1] `https://www.tiltbrush.com/`

Acknowledgements

Firstly, I express my sincere gratitude to my supervisor Prof. Anthony Steed for the support of my PhD study and research, for his motivation, patience, and vast knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better supervisor and mentor for my PhD study. Besides my supervisor, I would like to thank Dr Stuart James for his insightful comments and encouragement. His positive attitude spurred me to always improve myself and my research. My sincere thanks also go to Prof. Karol Myszkowski, who provided me with an opportunity to join his team during my secondment in Germany, and who gave access to the laboratory and research facilities. Without his precious support, it would not be possible to conduct this research. I thank my fellow lab-mates for the stimulating discussions, for the sleepless nights we were working together before deadlines, and for all the fun we shared in the last years. In particular, I am grateful to Alejandro for enlightening me the first glance of research and support me with inspiring ideas during all the PhD. I thank my best friend Mirco that supported me in the hard times and inspired me with enlightening chats. Last but not least, I would like to thank my family. Here in London with my beloved Ilaria and my adorable son Dante, born almost three years ago that always supported me unconditionally. In Italy, my parents Daniela, Vanni and my sister Anna for supporting me spiritually throughout this wonderful British experience and my life in general. And all my lovely cats, especially Jerry, that will be always in my heart.

Contents

List of Figures

Glossary

AlexNet CNN Model for image recognition, winner of 2012 ImageNet Large-Scale Visual Recognition Challenge. It is characterized of a very small number of convolutional layers and a final fully connected layers section.. 1

BoW Bag Of Words is a information model used mostly in natural language processsing in which text is represented by a set of its words.. 1

CNN Convolutional Neural Network, is a neural network inspired by visual cortex that is composed by convolutional and pooling layers in addition to rectified linear units that ensure non linearity. It is largely used in computer vision since demonstrated in last years high performance on image classification.. 1

CPU Central Processing Unit is the logic processor of a computer in which normally most calculations take place.. 1

DNN A deep neural network is an artificial neural network in which multiple hidden layers are between the input and output layers. They can model complex non-linear relationships.. 1

FCL Fully Connected Layer is a layer in a neural network in which each unit is connected with each unit of the following layer.. 1

FCN Fully Connected Network is a neural network composed by FCLs.. 1

Gf-HOG Gradient Field Histogram of Gradient is an adapted version of HOG specialized in Sketch Base Image Retrieval systems.. 1

GPU Graphics Processing Unit is a programmable logic processor specialized for rendering images, video for computer displays.. 1

GUI Graphical User Interface refers to a type of interface in which the user can interact with graphic objects displayed in a monitor like buttons, menus, windows.. 1

HCI Human Computer Interface is an area of research that focuses on interaction between users and computer. It can be considered an intersection of computer science, behavioral science, design.. 1

HMD Head Mounted Display is a wearable device, with a small display for one or both eyes. Different devices were launch in the market, and recently Oculus Rift gained lot of popularity.. 1

HMM Hidden Markov Model is a statistical model in which the system is a Markov chain without observed states.. 1

HOG Histogram of Gradient is a feature descriptor used for object detection. It uses distribution of local intensity gradients or edge directions.. 1

MOS Mean Opinion Score is a test for evaluating image quality that asks to user the quality of the perceived image using five level of magnitude.. 1

Pooling Pooling in CNN refers to a specific layer that reduce spatial size of the input. Pooling can be Average Pooling or Max pooling that takes respectively the average or a max of a values subset.. 1

PSB Princeton Shape Benchmark is a 3D Model repository used for classification and retrieval problems created in Priceton.. 1

ReLU Rectifier Linear Unit is an activation function used in convolutional neural networks, other options can be sigmoid function, hyperbolic tangent.. 1

RNN Recursive Neural Network is a deep learning model that apply the same weight recursively along the network architecture.. 1

SBIM Sketch Base Interface and Modeling is an interaction paradigm in which the user convey information to the system through sketches instead using the normal WIMP methods.. 1

SGD Stochastic Gradient Descent is a method used in backpropagation stage during training a neural network for minimize the error function.. 1

SIFT Scale Invariant Feature Transform is a compuiter vision algorithm that describe image features; it extract keypoints and with them computes descriptors.. 1

SSIM Structural Similarity Image Metric is a full reference metric based on perceived change in structural information; it is represented by an index calculated for all patches of the image. 1

SVR Supported Vector Regression is a supervised learning method used for regression analysis. 1

Unity3D Unity 3D is a game development platform for Windows and MacOS used for creating software based on 3d graphics.. 1

VGGNet CNN Model for image recognition, winner of 2014 ImageNet Large-Scale Visual Recognition Challenge. 1

VR Virtual Reality is a computer technology for generating realistic 3d images or sounds for simulating an environment and the user immersive experience in it.. 1

WIMP Windows Icons Menus Pointers refers to a style of interaction in which GUI components and pointers are the mechanism used by user to interface with the system.. 1

Chapter 1

Introduction

The rapid expansion and diversity of all forms of digital information is leading us into a period for computer science and human history that is named the Zettabyte Era [8]. In particular, with the rapid growth of interest in 3D modelling, repositories of 3D objects have ballooned in size [1, 9, 10]. These collections are particularly valuable to many fields including robotics for use in simulations to the real world, and computer vision in object detection and recognition, object pose evaluation and spatial 3D interpretation. The collections are constructed by both synthesising 3D models using modelling software – Computer-Aided Design (CAD); or by scanning real-world scenes with RGB or depth cameras where 3D objects can then be extracted in a post-processing 3D reconstruction stage. As these datasets are a source of interest for researchers in computer graphics and vision, the motivation for their growth or improvement is present and constant. On the other side, model collections curated by the game development community increased in popularity concerning the diffusion of frameworks such as Unity [11] or Unreal [12] with convenient asset stores. Although these datasets are not all publicly free, many companies permit 3D modellers to share their creations and perform searches among the whole collections.

For most software, the methods that deal with the retrieval of 3D objects are linked to interfaces that rely on text queries. This is still the standard despite Human-Computer Interaction researchers study and propose new paradigms for more intuitive interfaces (e.g. Sketch-based Image Modelling, Gesture-based User Interface, Vocal User Interface, [13, 14, 15, 16, 17]).

Although a text query may efficiently convey semantic labels, such as, chair, table, cup, it quickly becomes cumbersome and generally fails with complex visual characteristics, e.g. “chair with slatted back, iron arms and decorated legs”. Also, a text query can be influenced by a lot of variables that depend on the user: educational level, past experiences, mental inclination, as well as the current state of mind that compounds search performance difficulties. Furthermore, in many existing databases,

meta-data fields are incomplete, too generic, and sometimes totally missing. Illustrated queries can efficiently communicate such concepts and can be used in combination with other techniques to improve the results. Therefore Query-by-Example (QBE) methods have become a very active area of research. In QBE systems, the user typically provides an example in the form of an image or depicts elements of the object or scene they wish to retrieve. A search system then retrieves matching elements from the database. Typically an image is not to hand, thus sketch is a more useful form of QBE.

The sketch is an intuitive and familiar method to convey information that has two essential characteristics: firstly, it is compact, and it requires a short time to be generated, and secondly, it is persistent in time such as written words. A sketch can transmit shape information (2D or 3D) and colour at the same time. A sketch can suggest a category of an object or, with more details, can be a specific instance of a group of objects. On the other side, sketching comes with an expressive power only matched by a high degree of freedom, and this represents a problem. Interpreting a sketch becomes a difficult task if the user depiction is very abstract or avoid some characteristics. Moreover, this task becomes even more challenging if the sketch contains defects or exaggerations, an ordinary thing for this type of communication. For this reason, sketch input is not a widely spread method of interaction, and more traditional 2D interfaces are still preferred today. Text or 2D interface elements such as combo-box, list-box, palette, or sliders are used to indicate characteristics of an object belonging to its structure or colour appearance. Such functionalities are simple to understand and use individually, but the whole user interface may become complex and difficult to navigate when a large number of elements are required for the search task.

Recent years have also seen Virtual Reality (VR) gaining increasing interest with a new generation of devices both for PC and for gaming consoles. Their potentialities open a new era of immersed visualisation and interaction based on gestures, movements of head, and limbs. Therefore, either the translation of traditional interactions or the development of new interaction styles is demanded for such systems. The contemporary trends that embrace 3D content and this new form of interaction are the motivation for realising new paradigms and algorithms for managing tasks like searching between the available 3D data.

Machine learning algorithms have been studied by decades. They are based principally on extracting and use knowledge from the data. One of its subsets is deep learning with neural network models that can boast millions of parameters to be adjusted to accomplish a classification or regression task. These models gained increasing popularity for the outperforming results achieved in computer vision, natural language processing (NLP), automated speech recognition, reinforcement learning.

Numerous science fields have benefited from this revolution, as tasks considered intractable or very hard to solve with hand-crafted solutions now are accessible and affordable also for everyone, even without domain-specific knowledge.

1.1 Interaction and algorithms in mixed environments

Immersive environments are simulations located in a virtual digital space built around the user. The user's field of view is filled by this virtual environment, providing an experience of physical presence in a digital space. Virtual reality can bring immersion to the single user that wears the head-mounted display (HMD) and interacts with objects in the virtual environment through controllers. For an immersive experience, the objective is to involve the user's senses fully – sight, hearing, smell, taste and touch; however current technology facilitates sight hearing and, to some extent, touch. Virtual reality has aroused interest since its inception, but it is in the last five years that industry has developed a strategy conducive to its consolidated development. The release of commodity-level devices such as Oculus Rift and Touch [18], or HTC Vive [19] have laid the foundations of consumer and business interest in VR. Where the estimated worldwide VR market forecast by 2023 is 34.5 billion pounds (source: Greenlight Insights [20]). This technological revolution that touches both the visualisation and the interaction becomes fertile ground for the development of new paradigms and methodologies in which the user is the primary experimenter. VR controllers and finger tracking devices (such as Leap Motion [21]) can be paired with VR display devices and provide interaction with a higher number of degrees of freedom (DoF) than the personal computer's mouse or console's game controller. DoF indicates the number of movements (rotations and translations) a rigid object can do in a three-dimensional space. For a rigid body, the maximum number of DoF is six: three rotations and three translations. 6-DoF devices and two hands to track such as Vive or Oculus devices, increase freedom of interaction and require better recognition algorithms. Fortunately, in recent years, an order of new algorithms has shown to obtain excellent results precisely - but not only - in the context of classification problems: deep learning algorithms.

Our work provides the user with a novel sketch-based interface in an immersed environment for retrieval tasks. We analyse different types of sketch interactions, spanning from 2D to 3D modes. In addition, we will show different solutions for image feature extraction that represent the core algorithms for interpreting the sketches. Moreover, we explore a multi-modal sketch and voice-based interface to tackle the most challenging objects when searched. Finally, we apply the same category of image algorithms to visual perception so to show how other fields of computer graphics can benefit from

the same procedures used in human-computer interaction. These works are innovative and show a distinguished accuracy in either sketch interaction for the immersed environment and visual perception.

1.2 Deep Learning Applied to the Visualisation Pipeline

The visualisation pipeline (Figure 1.1) is a metaphor that illustrates a sequence of transformations that starts from the data and finishes with the visual representation of it. It embodies mechanisms such as data loading, data filtering, rendering, and over the years, different implementations have been proposed [22]. Being the basis for computer graphics libraries that are used for data visualisation and manipulation, user interaction can be present at different stages of the visualisation pipeline [23]. Moreover, at the end of the visualisation pipeline, visual quality metrics gained importance in the context of improving the perception of the displayed data.

In computer graphics, machine learning (ML) has impacted deeply, with many classic problems handled with data-driven methods. Deep networks are the state-of-the-arts improving considerably traditional hand-crafting algorithms. Colourisation [24], real-time rendering [25], BDRF estimation [26], mesh segmentation [27], animation [28], sketch simplification [29], denoising [30], are some examples where deep learning models provide outperforming results.

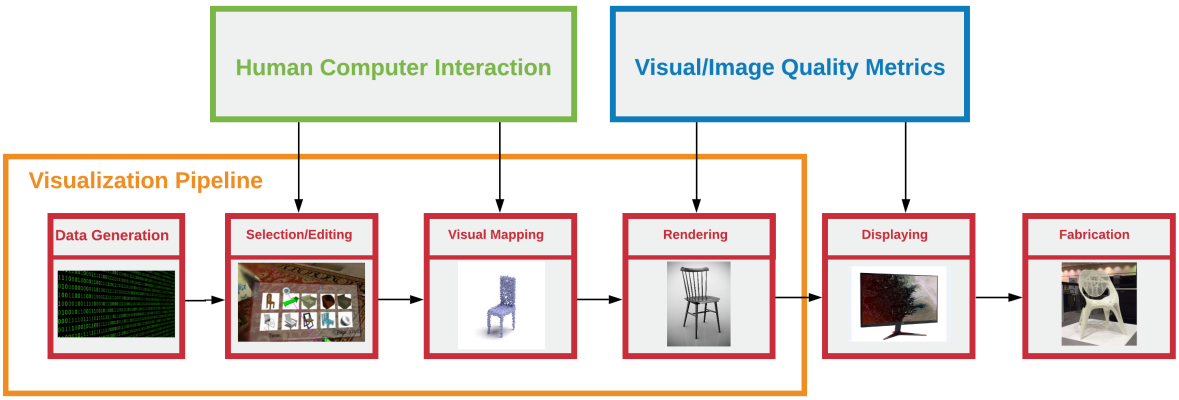


Figure 1.1: A possible pipeline (red boxes) contains the visualisation pipeline, and it is extended with Display and Fabrication blocks. Human-computer interaction and visual metrics supported by deep learning algorithms are the two areas on which this thesis is focused.

1.2.1 Deep Learning and Human Computer Interaction

Machine learning is revolutionising the way we are interacting with computers. Human-Computer interaction is a multidisciplinary science, and different areas benefit from deep learning algorithms. For example, natural language processing (NLP) is a technology that allows a system to understand human speech, improving the classic limited voice command interface [31], and at the same time,

empower chat-bots with AI-based algorithm [32]. NLP with voice recognition is the core technology used in voice assistants like Alexa (Amazon), Siri (Apple), Cortana (Microsoft). Thanks to progress in deep learning, some computer vision tasks such as gesture recognition [33] or eye-tracking [34] achieved significant results in terms of better accuracy in classification and noise reduction.

1.2.2 Deep Learning and Visual Quality Metrics

In the last decade, visual quality metrics gained interest in the role of evaluating algorithms such as for lighting, imaging, and rendering. Moreover, 3D Meshes are subjected to be assessed by algorithms that measure their visual quality. The state-of-the-art algorithms are principally hand-crafted solutions such as HDR-VDP [35] or [36] for the 3D meshes. Only recently, deep learning methods were used to evaluate the quality of images [37], [38], [39], paving the way for future works based on data-driven algorithms.

1.3 Contributions

This thesis contributes novel deep learning-based algorithms and applications in a mixed environment for searching and analysing data using interaction to generate textures as input. In addition, we study the effect of multiple modalities, specifically the combination of sketch-based interaction with spoken one. Finally, we exploit the convolutional neural network able to detect image differences in the context of visual perception.

We propose several interaction methods and algorithms based on Sketch-based Image Retrieval (SBIR) or the broader Sketch-based Retrieval (SBR), but with the additional difficulty of introducing the mid-air sketching action in an immersive environment achieved with virtual reality devices. Through the years in computer graphics and human-computer interaction, sketch-based content retrieval gained much interest but applied in a virtual environment with the 3D sketch as input is a substantially unexplored research branch. We delineate our contributions to SBR in VR in Section 1.3.1.

Therefore, we also study and analyses a multi-modal interface with two input channels: sketch and vocal data. We compare the results of the retrieval task exploiting both the individual interactions and the combination. Although sketch input shows valuable performances, some specific cases are more challenging, and the outcomes are less accurate. In this work, we focus on the generation of a database that enhances such complications, and we tackle them, formalising the concept of a query by example for sketch and voice. Therefore, we show how it is possible to combine sketch and vocal queries with improving performance in a retrieval task. We outline our contributions to sketch and voice multi-modal interaction in Section 1.3.2.

Finally, we apply a modified version of the previous studies' algorithms mentioned in a subarea of visual perception. A large number of applications in graphics and imaging can benefit from knowing whether introduced changes in images are visible to the human eye. For example, when visualising a complex scene, we may wish to select downsampled textures indistinguishable from the full-resolution textures. This work shows that the same class of algorithms that exploited sketch feature extraction can be used successfully to define a visual metric that discovers perceptible differences between images. We summarise our contributions to visual metrics in Section 1.3.3.

1.3.1 Sketch-based Model Retrieval in an Immersive Environment

This thesis contributes to new techniques for immersive environment interaction using virtual reality. Firstly, we compare different sketch modalities in virtual reality to determine which interaction achieves the best accuracy and provides the best user experience. We then analyse different solutions for extracting features from the sketches, demonstrating that deep learning algorithm achieves better performances in term of accuracy. Therefore, we implement a novel interaction modality in an immersive environment that uses both mid-air sketching and 3D item selection and a deep-learning model as back-end. This system provides the user with an efficient mechanism for 3D object retrieval. The hypotheses were the following:

[H1] 3D sketch can be used as a form of visual input in an immersive environment that helps to create a valuable descriptor of a 3D model. Users can interact with a virtual environment through body movements such as gestures or sketch using their hands. These inputs can be visualised and even be persistent in the scene. In the specific case of sketches, the user can depict using multiple colours, can change the size of the strokes and may apply some linear transformations to them or their copies like rotation or translation. To demonstrate the validity of our idea, we test it with the common task of searching in a database. Therefore, we define from this general argument three detailed research hypotheses:

[H1.1] 3D sketch is an effective interaction to depict information in an immersive environment with the aim to retrieve models from a large collection.

Nowadays, 3D model repositories are common and include a large number of items. Whereas models are tagged with keywords that describe their shape or colour attributes, it is difficult to depict all the characteristics of an object. In this context, sketch input represents a valid alternative that belongs to query-by-example methods. Sketch-based retrieval systems are widely explored in 2D

desktop solutions for both 2D and 3D data. The use of the sketch as query does not require additional information in the database and so it results free from issues like missing or incomplete text fields or tag. We design and implement a sketch-based retrieval system based on multi-view camera snapshots in virtual reality that aims to describe in the form of oversimplified visual depictions a 3D model. Sketching within the virtual reality raises several questions for the style of interaction, be it to replicate the familiar real world, sketching in the air, or trying to mix the two. In this study, we look at different methods of interaction, i.e., 2D physical tablet, a 2D Virtual Tablet, and 3D 'air' drawing, with the aim to understand how people naturally perform sketching within a virtual environment. We utilise a mixture of physical devices and virtual interfaces based on 2D projections. We frame the problem of retrieval and find real immersive 3D sketching as a more intuitive form of expression of a sketched query. We pose this as a retrieval problem, but it theoretically can be extrapolated to other tasks.

[H1.2] Deep learning models are more effective for extracting features than well-known detectors and descriptors when applied to sketches multi view 2d-sketch supervision.

A 3D sketch is a time-evolving sequence of coloured strokes that can be consecutive or completely detached with arbitrary length. With this kind of time-variant structure, a feature analysis in the context of retrieval task becomes extremely difficult for one main reason: a sketch represents an idea in the mind of its author. A sketch is typically done quickly in order to convey key features that the user imagines. Tackling the issue of how to predict what a sketch is representing can be essentially translated into predicting what an image is showing. Our system is specifically designed to collect images from different angles of interest of the sketch and use them as input for the feature extraction stage. We deal with the time-varying nature of the sketch allowing the user to interrogate the system in each possible growth stage of the sketch. Common techniques such as scale-invariant feature transform (SIFT) [40], Histogram of Gradients(HoG) [41], Gradient Field Histogram of Gradients (GF-HoG) [42] generate vector data that aim to describe the visual features of the contents in images. These descriptors are also achieved by the first stage of convolutional neural networks (deep learning algorithms). We test this hypothesis through an internal study over a set of sketches that refers to specific 3D objects measuring the accuracy of the outcomes.

[H1.3] Feature descriptors generated by the combination of sketch and 3D model in the context of an interactive loop, are an improvement in terms of accuracy.

Sketch alone leads to results that depend on the drawing ability of the user. Frequently it

degenerates to depicting the structure of the object and required colour. The iconic nature of the sketch is perfect when searching a class of objects among a set of different categories (e.g., searching for a chair among the set of furniture or an aeroplane in the collection of transport vehicles). However, the sketch shows some problems when the search regards a specific instance of an object among the same class of objects (search a tall chair among a chairs collection). When performing a sketch session, the user finds it challenging to fill surfaces with 3D strokes because it is a time-wasting activity, and it is common for users to forget some regions. Therefore, some related attributes that are considered coarse-grained such as the volume occupied by the chair are difficult to describe precisely only with the sketch. In addition, in a chair collection, the parts of each chair can always be considered present such as the seat, usually present, the legs or the back, or optional such as the arms. Despite their presence or not, each of them has more specific characteristics that depict a style, a structure, or a texture. For the above reasons, to describe each chair, each component needs to be delineated by a descriptor that considers both the coarse and fine-grained features. We develop a solution to this problem enriching the user interaction with the functionality of generating such a descriptor, not only using the sketch but using a combination between the sketch and the 3D model present in the scene. This approach increases the possibilities for the user to navigate the collection through the combined visual features of sketch and 3D objects.

1.3.2 Multi-modal interaction in virtual reality

Multiple modalities frequently provide the user with different and complementary tools for achieving a given task. When two different modalities are applicable for the same task, the system has redundant modalities. We are interested in exploring how a redundant system with sketch and speech interaction can improve the accuracy of the result for the search task and how the user experience is affected by this combination. Sketch and speech are two types of input that share the advantage of being natural and intuitive when conveying information. Sketch gives the user the expressive power of visual representation with the direct manipulation of the appearance. Speech is a form of interaction that takes advantage of the semantic power that language can give.

We describe and evaluate a novel multi-modal interface within a virtual environment for searching 3D model databases that combine 3D sketch and voice input as queries. The hypotheses were the following:

[H2]The task of searching for 3D objects can benefit from speech, as an additional semantic channel, in a redundant multi-modal system with sketch interaction. In the previous section, we

outlined what the issues of searching by pure sketch are. With the additional features coming from the model, we notice an improvement in terms of accuracy. However, the chair collection present in ShapeNet does not cover all the possible combinations of shapes and colours, and it is hard to determine potential bias coming from the dataset. On the other hand, a dataset that includes all the permutations between chair parts and colours and textures would be intractable because of the dimensions. To make an extensive analysis of the efficiency of the sketch method, we created a smaller version of the fully variational dataset, discovering that search-by-sketch accuracy drops dramatically. To tackle this problem, we design a two input channel workflow, with additional speech interaction to improve the efficiency of the search. However, speech interaction can not be included in the system without studying some aspects of natural language processing (such as readability or richness). In addition, a redundant multi-modal system needs to integrate such interactions minimising possible issues such as input clashing and unbalanced search processing. Therefore, we detail three research hypotheses:

[H2.1] Optimised lexical readability for textual information obtained from speech interaction can improve the efficiency in describing an object for retrieval. In the context of live interaction, textual object description is the output produced by the user that orally describes the object. Having good readability and richness of the textual information is crucial when searching an object by describing it. While people use the language daily to exchange information of various kinds, NLP models - such as recurrent neural networks - are trained to predict or classifying specific text based on some subset of corpora. A well-known problem in recurrent models is the gradual decline in the importance of the words that precede the current one. Also, there is the possibility that contradictory information happens within the text. An autonomous system that takes speech in input and produces a set of possible target objects in output is composed of different stages. Firstly the speech needs to be recognised and translated into textual information. This stage can introduce errors that can affect the system output. Secondly, a tokenisation and lemmatisation stage needs to be performed to improve readability. A final stage of semantic interpretation of the pre-processed text has to produce probabilities associated with each model of the database and select the ones with more likelihood. Such a system would be difficult to debug and control for the different errors that each stage can add. Therefore, we design an experimenter-in-the-loop test to determine the appropriate length of the query that a user can say to describe an object. Therefore we can fix the number of relevant words of the query managed by the experimenter to build a descriptor. This descriptor will

be used to determine which models will be presented to the user for the final selection during the search.

[H2.2] A multi-modal system including sketch and speech, to work properly, needs a formal query definition and a specific interaction pipeline. Multi-Modal interaction with sketch and speech requires a formal definition for the query and the interaction workflow. The search session can be composed of multiple sketch queries and speech queries. The output of any query must be compatible with the input of any query, creating a valid sequence. In addition, the speech query needs to avoid an eventual bias added by the experimenter. We make a formal definition and implementation for both the query types. Therefore we illustrate the workflow to create the sequence of queries and analyse the different user strategies that emerge.

[H2.3] The combination of voice and sketch interaction improves the search in an immersive context when compared to individual techniques.

We run a user test that compares the efficiency for three different methods: pure voice interaction, genuine 3D sketch interaction, 3D sketch, and voice combined interaction. We designed two user experiments, showing the optimal number of words that the user needs to use during a single vocal query and demonstrate that the combination of 3D sketch and speech interface achieves better results than the other modalities taken individually. Using this redundant system, users can overcome some difficulties in retrieving models from large datasets, bringing together visual queries from a 3D environment and the semantic component coming from voice communication.

1.3.3 Data-driven visual metrics

Many applications in graphics and imaging benefit from knowing how introduced changes in images impact human visual perception. For example, we may wish to select textures of possibly small resolution that are not introducing visible artefacts and are perceived without distortions when rendered in a complex scene. State-of-the-art algorithms (such as Structural Similarity Index Metric [5]) that predict differences between images do not achieve satisfactory results. Our work moves firstly in the direction of creating the largest collection of computer graphics-based distorted images. Secondly, we create a deep learning model crafting a probabilistic loss function that can predict visible distortions. My personal contribution is summarised in the following points:

1. writing part of the software used in user experiment

2. CNN model implementation, in particular fully convolutional architecture and the statistical loss

function

3. analysis of super-resolution application.

[H3] Deep learning algorithms can improve image difference prediction in the context of computer graphics. Putting two images in front of a user and ask to identify the differences is a very popular game. Predicting the visible differences between the two images looks to be the same task, but it is not. In the first case, the user needs to search the difference and therefore detects it. In the second case, the prediction does not need to consider any searching component over the image and states if the difference is visible or not. Deep learning algorithms show their efficiency in many fields of computer science, and we apply them in the context of visual perception by creating a model that marks all the locations in the image that are perceived as different from a reference image. Thus, we outline the following detailed research hypotheses:

[H3.1] A statistical loss function applied to a CNN-based model improves the accuracy in visible differences detection task.

To create a robust predictor of visible distortions, we collect a large dataset with locally marked distortion, and we calibrate popular visible difference metrics achieving performance improvement. Secondly, we use the dataset to train a novel metric based on a CNN architecture. During the training phase, we use a statistical loss function that excludes the search component present in the user experiment results. This solution improves the prediction accuracy over existing metrics by a substantial factor.

[H3.2] A fully convoluted model boosts its accuracy in predicting visible artefacts using a training set that includes an extensive collection of annotated distortions in computer graphic. Overfitting is one of the most common problems when training a neural network. Overfitting happens when the model is not able to generalise the prediction to a problem outside the training set, and conversely, can predict all the elements of the training set successfully. Different solutions can be adopted to overcome this issue. To begin, extending the training dataset and then reducing the number of the parameters of the model. We apply both these techniques to improve accuracy over state-of-the-art algorithms avoiding overfitting. Therefore, we increase the training set by data augmentation using rotation and flipping operation over the images. Thus, we edit the two-branch neural network architecture with the fully connected layer in a fully convoluted network with residues to reduce the

number of parameters drastically.

1.4 Structure of this Thesis

We describe the chapter structure of the rest of this thesis, and we outline the main contributions.

Chapter 2 — Literature Review

This chapter contains an exhaustive literature survey of the areas of interest for this research: sketch-based retrieval, sketch in mixed realities, multiple modalities of interaction, and visual metrics.

Chapter 3 — 3D Sketching for Interactive Model Retrieval in Virtual Reality

We present an experiment where we focus on supporting the user in designing the virtual environment around them by enhancing sketch-based interfaces with a supporting system for interactive model retrieval. Through sketching, an immersed user can query a database containing detailed 3D models and replace them in the virtual environment. To understand supportive sketching within a virtual environment, we compare different methods of sketch interaction: 3D mid-air sketching, 2D sketching on a virtual tablet, 2D sketching on a fixed virtual whiteboard, and 2D sketching on a real tablet. Therefore we analyse the performance of different descriptors to understand which one achieves the best accuracy. We then improve the system providing the user with a novel way to combine sketch and 3D model in a single query.

Chapter 4 — Multimodal approach fusing sketch and speech in Virtual Environment

We introduce a redundant multimodal system that makes use of speech and sketches simultaneously for retrieval purposes. We formalise the definition of the query, generate a database that enhances the difficulties showed by the sketch retrieval system, and design an experimenter-in-the-loop user test that processes speech input providing the semantic information of text description. We run a test for determining that such a multimodal system achieves more accurate results than individual interactions.

Chapter 5 — Dataset and Metrics for predicting visible differences

We apply the same algorithmic procedure of extracting visual features for sketches in the field of visual perception. After collecting an extensive database of images with visible distortions and relative references, we train state-of-the-art visual metrics to predict artefacts on distorted images. Therefore we introduce a novel predictor model that, after appropriate training, outperforms the previous visual metrics and also a trained version of them.

Chapter 6 — Conclusion

We evaluate the contributions related to the respective research hypotheses (H1-3) described in this chapter.

Chapter 2

Literature Review

2.1 Introduction

In this chapter, we describe prior work on 3D object retrieval systems, focusing in particular on convolutional neural network (CNN) implementations that currently represent the state of the art for feature extraction algorithms. We present recent studies in which sketch-based interaction is used in combination with CNNs. We analyse recent works on spoken interaction starting from command-based ones, followed by more complex implementations in virtual reality scenarios. In the last section of the chapter, we describe visual metrics that define the modalities of prediction of visibility differences between images and creating a database of distortions and annotations. Based on the state-of-the-art algorithms described in the literature review, we aim to design and implement novel interaction paradigms, to improve accuracy in tasks such as searching in a database or correcting data.

2.2 Sketch-based Retrieval

Sketching is a direct way for people to convey information. Eitz et al. [43] describes how people depict objects and how both humans and computers recognise sketches. The primary supposition is that sketches approximate the real-world object. On the other hand, since the average user is not an artist, the subjective depiction of an object can be representative and include simplifications of it. We explore the implications of this for both retrieval and interaction for both 2D (Image) and 3D domains.

2.2.1 Sketch-based Image Retrieval

Identifying and associating a sketch with a specific object in an image represents a hard challenge. However, it is an attractive strategy because the use of sketch interaction is an opportunity to broaden the user base to those who are unfamiliar with complex interactive editing systems.

Various methods for retrieving images from sketches have been developed. These systems are

referred to as sketch-based image retrieval (SBIR) systems [44]. SBIR techniques can be classified into two classes: blob-based techniques that focus the attention on features such as shape, colour or texture, and contour-based techniques that describe the image using curves and lines. Techniques belonging to the blob-based SBIR class try to describe image through descriptors such as QBIC [45] which use separately colour, texture and shape or [14] which uses topology models. Contour-based techniques include elastic matching [46] and grid and interest points such as edge points [47].

In recent years researchers have applied machine learning algorithms to SBIR. SketchANet [4] is a simple neural network based on Alexnet that performs sketch recognition. Qi et al. [48] introduce a siamese CNN which aims to measure the compatibility between image edge-map and sketch used as CNN inputs. Bui et al. [49] did a review of different triplet CNN architectures for evaluating the similarity between pictures and sketches, focusing on the capacity to generalise between object classes. Triplet architectures [50, 51] have attracted increasing attention for the relationship of the three branches when processing the loss function: firstly the anchor branch (modelling the reference object), secondly a branch which models positive examples and thirdly a branch that deals with negative examples.

A strategy to improve the performance of image retrieval systems is to put the user 'in the loop' and take advantage of iterative refinement. This technique is called relevance feedback in information retrieval and was introduced in Content-Based Retrieval by Sciascio et al. [52]. Several applications based on interactive sketch systems have been created. For example, Shadow Draw from Leeet al. [53], iCanDraw [54], Sketch-to-Collage [55] and CALI system from Fonseca et al. [56].

2.2.2 3D Sketch-based Retrieval and Interaction

Finding features that represent 3D objects is a unique challenge in the retrieval domain. One of the most important cues in object recognition is a 3D geometric shape. Generally, the sketch is a simplified representation of an object, and this gap between the real and sketched shape can hamper the classification process. In addition, before sketch interpretation, a simplification process of the stroke can be taken for avoiding noisy samples [57] since both the tracking device and user generate noise during sketch acquisition.

In recent years, to depict a 3D model, researchers have proposed two types of descriptors: model-based and view-based.

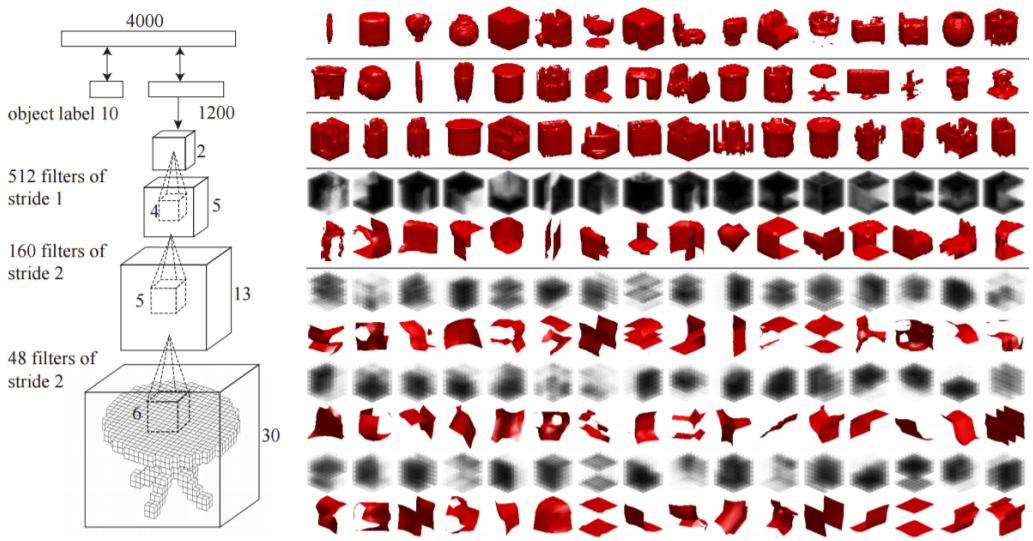


Figure 2.1: The architecture of the model of 3D ShapeNets on the left. On the right, the grey images are the average taken from 100 training examples for visual features extracted from the neurons of the model (both Figures from [1])

2.2.2.1 Model-based descriptors

Measuring similarities between 3D models is a hard problem. Object models can differ in shape, colour and orientation in 3D space, making the definition of a similarity measure challenging. Different categories of descriptors were created to overcome this challenge: geometric moment, surface distribution and volumetric descriptors. Geometric moment [58] is a class of topology invariant similarity methods based on vector coefficient extracted by a shape decomposition under specific basis. Surface distribution [59] tries to measure the global properties through a shape distribution achieved by sampling a shape function and in this way reduces a shape comparison to a simpler distribution comparison. Volumetric descriptors [60] combine shape distributions with barycentroid potential for achieving a more robust pose and topology invariant similarity. Despite the extensive research on descriptors that allows extracting shape characteristics, only with the advent of deep learning architectures such as Restricted Boltzmann Machines (RBM), Deep Belief Networks (DBN) and Deep Boltzmann Machines (DBM), and in particular convolutional neural network (CNN) [61] have achieved a relevant improvement of outcomes in object recognition. Wu [1] recently proposed a method to represent a 3D object through the distribution of binary variables in a volumetric grid, and use of Convolutional Deep Belief Networks to extract features and recognise them (as Shown in Figure 2.1).

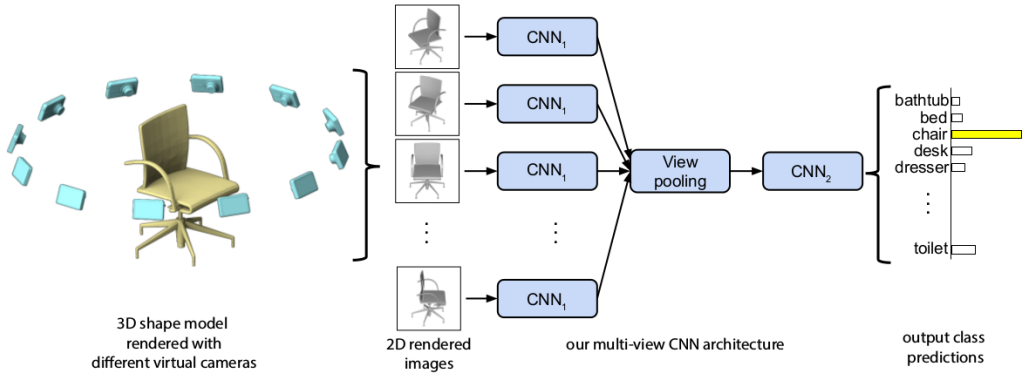


Figure 2.2: A multiview model takes advantage from images taken from different point of views that are processed by CNN in parallel. After a pooling layer an additional CNN performs the classification (Figure from [2]).

2.2.2.2 View-based descriptors

View-based descriptors use 2D projections of the objects from different points of view. Since a large amount of data can be collected in this way, these methods outperform model-based descriptor approaches. Ansary et al. [62] introduce a model-index technique for 3D objects that make uses of 2D views. It uses a probabilistic Bayesian method for 3D model retrieval. Alternatively, Su et al. [2] present a framework using view-based descriptors, creating 12 views for each object that feed a first CNN for feature extraction, and after a pooling stage, the results are passed to another CNN for achieving a compact shape descriptor (as shown in Figu 2.2). Similarly, Leng et al. [63] proposed a 3DCNN that manages multiple views and considers possible interactions between them. In a pre-processing stage a sorting algorithm, which takes into consideration the angles and positions, prepares three different sets of viewpoints and the network is fed with them at the same time. This is a different approach from the classic one which uses only one view at a time, and it confers stability during the training stage.

Eitz et al. [3] describes a 3D shape search engine without considering deep learning models, where local descriptors are generated through visual vocabulary represented by a collection of visual terms. This concept is taken from natural language processing (bag of words). This architecture is shown in Figure 2.3.

Li et al. [64] elaborates a technique that combines two components: an adaptive view clustering algorithm that selects representative views of the 3D model, and a sketch-based approach that compensates the difference between the iconic representation of the object given by sketch depiction and the detailed appearance of the same object.

Our method uses view-based descriptors, rather than model-based descriptors because they have

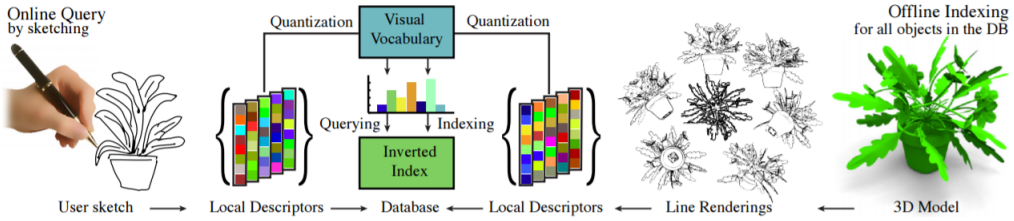


Figure 2.3: 2D Sketch to 3D object pipeline that shows descriptors created from sketches and from 3D models and compared with a histogram of visual vocabulary (Figure from [3])

demonstrated more practical utility on similar problems (see Section 3.3.2.2).

Recent studies combine data achieved by sketches and additional input to increase accuracy, or infer information from the relationship with other objects in the scene. Funkhauser et al. [65] proposed a combination of sketch and text query to identify 3D objects. They showed that the combination of the two methods results in better accuracy of the results. Shin and Igarashi [66] with Magic Canvas provided the user with a system for 3D scene construction using sketches, based on sketch-object similarity. In addition, the system determines the position and orientation of the object according to the sketch. Xu et al. [67] with Sketch2Scene proposed a novel framework for scene modelling through sketch drawing that also suggests the placement of the objects via the functional and spatial relationship between objects.

Critically these methods have generally used 2D sketches. Our system allows the user to sketch in 3D.

2.2.2.3 3D sketching

3D sketch-based model retrieval has gained significant attention in recent years, boosted by the increasing number of devices available for the consumer VR market. Li et al. [68] made a comparative evaluation of different 3D sketch-based model retrieval algorithms showing that CNN, in combination with edge or point sketches, achieved the best accuracy.

Ye et al. [69] described CNN-SBR, a CNN architecture based on SketchANet [4] and trained with TU Berlin dataset [43]. Using data augmentation to prevent overfitting, they showed a considerable improvement in comparison to non-learning based and other learning-based algorithms. SketchANet used DNN algorithms to extract features from simplified and deformed versions of the sketches and fuse the results through Bayesian process to further improve recognition performance(as shown in Figure 2.4).

Considering alternative uses of sketch interaction within a 3D context, Wang et al. [70] present a minimalist approach in terms of view-based descriptors. They generate only two views for the entire

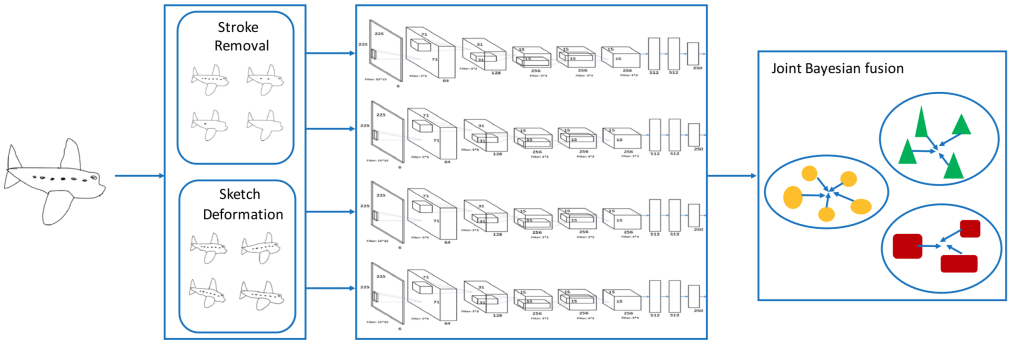


Figure 2.4: Overview of the pipeline for Sketch-A-Net (Figure from [4]) that is the base of CNN-SBR.

dataset and train a Siamese CNN with the views and the sketches. Nishida et al. [71] proposed a novel method to design buildings from sketches of different parts of them. The user sketches few strokes of the current object, and through a pre-trained CNN for that specific object type, the system is able to procedurally retrieve the correct grammar snippet and select the most similar one. The final step of the process is to combine all the snippets in a unique grammar of the building just created.

2.2.2.4 Immersive Sketching

Immersive sketch-based modelling has gained increasing attention over the years. A very early example is Clark's 3D modelling system for a head-mounted display [72]. The system of Butterworth et al. [73] supported several geometric modelling features, including freehand operations. More recently many immersive 3D modelling systems have exploited freehand sketching such as BLUI [74], CavePainting [15], Drawing on Air [75], FreeDrawer [76], Holosketch [77] and Surface Drawing [78]. Very recently, applications for consumer virtual reality systems such as Tiltbrush from Google and Quill from Facebook have raised awareness of sketching for content development. The most similar work to ours is the system Air Sketching for Object Retrieval [79]. This combines 3D sketch and a search engine based on the spherical harmonic descriptor. Our system uses a different type of lightweight sketching over basic models and a view-based descriptor. Another similar system is designed by Li et al. [80]: a content retrieval system that can benefit from sketch-based interaction using a Microsoft Kinect. Possibly because the sketches are relatively crude, they focus on distinguishing between classes of object. We focus on precise sketching to distinguish between similar objects in a large class of objects. Also, we enable sketching over existing models, rather than sketching from scratch (see Section 3.4.4.2).

2.3 Sketch in Mixing Realities

2.3.1 3D Sketching in AR/VR

Rather than transforming 2D sketches into 3D representations, immersive modelling environments allow users to sketch and design 3D content using mid-air interaction techniques. Such techniques utilising 3D space are intuitive to learn regardless of the user's expertise with the VR system [81]. A very early example of such an approach called HoloSketch [82] combined head-tracked stereoscopic shutter glasses and a CRT monitor with a six-axis mouse or *wand* for mid-air freehand drawing.

While 3D freehand drawing can support expert 3D artists [83] and shows improving accuracy and uniformity of sketched objects over time [84], interaction techniques for sketching in 3D are still limited in terms of accuracy and user fatigue. Realising an intended stroke during sketching is firstly bound by the user's perception of depth [85]. When users are unable to determine their active drawing location and spatial relationships with other contents inside the scene, sketching errors occur. Additionally, the added complexity of 3D compared to 2D sketching causes a higher cognitive load and increasingly taxes the sensorimotor system [84]. Lastly, as the absence of a physical resting surface leads to fatigue, accuracy is negatively influenced over time [85].

3D freehand sketching approaches can be further classified in categories that highlight the motivation as well as the limitation of such type of interaction: novel metaphors, sketch beautification, and both virtual and physical support surfaces.

Sketch metaphors Wacker et al. [86] designed and implemented a tool called ARPen, whose real-world position is tracked by a smartphone app, that lets the user interact via mid-air sketch. In addition, they evaluated through a user study different techniques to select and move virtual objects with such a tool.

Sketch beautification Feng et al. [87] exploited a beautification algorithm that analyses raw sketches and beautifies them to express user's intent more accurately. The beautification process rebuilds a sketch after segmentation and creates a beautified sketch using the segments' kinematic features. This mechanism resulted in an intuitive gesture-based interface that provides designers with mid-air sketching tools, without the need of any physical device.

Shanka et al. [88] designed a screenless 3D sketching system that uses RGB-D camera tracking the user input and applies a beautification scheme on the 3D sketch to remove the noise coming from the apparatus. The entire process of beautification involves data refinement, segmentation, segments classification. In the final step, they fit each segment into a geometric primitive.

Virtual surfaces In recent years in VR environments, the use of 2D surfaces has been explored to replace the inaccuracy brought by 3D tracking. Despite the lack of one dimension, some tasks are suited for a 2D device such as terrain editing [89] , user interface editing [90] or even object selection and manipulation [91]. Arora et al. developed SymbiosisSketch AR [92] that combines 3D drawing in mid-air with freehand drawing on a surface. They equipped the user with an Hololens, a tracked stylus, and a tablet and created a hybrid sketching system suitable for professional designers and amateurs. Machuca et al. [93] supported 3D immersive sketching using multiple 2D planes to create 3D VR drawings.

Physical surfaces Because a virtually rendered tablet in VR can not provide the same latency-free response that a 2D tool can have, different attempts of integrating a real physical 2D tablet have been made. Arora compared traditional sketching on a physical surface to sketching in VR, with and without a physical surface to rest the stylus on [85]. Additionally, they investigated visual guidance techniques to devise a set of design guidelines. Wacker et al. [94] studied how accurately visual lines and concave/convex surfaces let users draw 3D shapes attached to physical virtual objects in AR and Dorta [95] draws on virtual planes using tracked tablets.

2.3.2 Feedback for Virtual Reality

To ensure user interactions in VR remain intuitive, interactive VR needs to include feedback mechanisms in line with the visual stimulation inside the environment [96]. Haptic feedback allows users to experience object properties and aids in planning and executing actions to interact in a safe, efficient, and precise way. Research distinguishes three basic types of haptic feedback for VR, namely active, passive, and mixed approaches.

The PHANToM device [97] is an early example of an active haptic feedback approach. An actuated stylus can exert forces onto the user, rendering an impression of the tactile properties of the virtual environment. Sketch-based interfaces benefit from the addition of active feedback provided by the PHANToM device during sketch creation and editing [98]. More recently, similar work considered the addition of virtual canvas elements during stylus-based While users still faced fatigue during mid-air interaction, force feedback partially provided compensation. In passive haptic feedback approaches, physical objects are spatially registered to virtual objects to provide tactile feedback in IVEs [99]. When reaching out to touch a virtual object, the user touches a corresponding physical prop to experience appropriate feedback. One example considers a golf putting task, where high-fidelity passive haptic feedback are shown to improve performance [100]. Mixed techniques include dynamic passive haptic feedback [101] where active and passive haptic feedback elements are combined by equipping passive

proxies with actuating elements. The aim for this approach is to dynamically communicate object properties such as weight distribution [101] or material perception [102, 103].

2.4 Speech in Multimodal Interaction

As our work investigates the impact of multimodal interaction in computer-aided retrieval, in this section, we start with an overview of how multimodal interaction can improve the user experience (see section 2.4.1). Later we examine sketch or gesture and speech as human-computer interactions, focusing on the mixed-use in 2D desktop ambient (see section 2.4.2) then on how each separately is introduced in Virtual Environments (see section 2.4.3 and 2.4.4) Finally, we describe how their combination impacts on the interaction in Virtual Reality.

2.4.1 Multimodal Interaction

In recent years, studies have focused on Human-Centered HCI (HCHCI). HCHCI is a process that targets the person's needs relative to what is needed to complete a task. Many modalities can be exploited during the interaction process instead of canonical use of an interface. Systems created with this feature are called Many-Modalities HCI (MMHCI) and contribute decisively to the development of HCHCI. In the last 20 years, many techniques have been extensively studied such as face detection, gesture identification, motion analysis, vision or speech recognition, eye gazing, handwriting and sketching, and PDA or mobile interactions. It has also been shown that well-designed multimodal interfaces increase the efficiency and naturalness of HCI. MMCHI is a field that brings together different areas and disciplines as the computer vision or natural language process. The key challenge here is to merge information from different sources to obtain a continuous coherent flow of data to be interpreted.

Speech and gesture were combined in Bolt's Media Room (1980) [13]. It is one of the first examples of multimodal system and the user can interact with gaze, gesture and speech directly manipulating the graphics. Excellent results have been obtained with systems where the vocal channel is merged with another visual channel such as in Quickset [104] where voice interaction is merged with handwriting, or in BattleView [105] which combines voice interaction and gestures. In pervasive computing, the fusion between computing speech, expressions, and gestures has been implemented in Smart Rooms [106]. In addition, there is no shortage of examples where more than two modes are merged, such as in INTERACT [107] where face expression, gesture, audio, and eye gaze are fused. Many disciplines benefit from the improvement of multimodal interfaces such as biomedical where Multisense [108] allows planning of hip surgery by merging information from haptics, sight, and touch.

Other projects such as MUST [109], SmartKom [110], and Imix [111] merge speech and pen pointing.

Cohen et al. [112] demonstrated that while the gesture was ideal for direct manipulation, the speech was more suitable for descriptions, and combining them was even more efficient. As stated before, designing a good multimodal interface is challenging, and errors in this phase could impact the final experience as happens in Kay et al. [113]. That system implements speech and draw interface, where the movement of the cursor is controlled by a time-consuming command and control vocal interaction.

Each project adopts a specific way to fuse information between different channels. There is no standard classification for information fusion. We can distinguish a fusion of multimodal or fusion of context. Information fusion can be categorised according to the architecture, to the input, data method, process method, or data source. In MMHCI, fusion is categorised as data fusion, feature fusion, or decision fusion. Data fusion happens when the data is of the same type as in multi-sensor fusion. Feature fusion that is the most common method in MMHCI refers to modalities highly coupled, such as audio or video. This kind of method includes, for example, weighting average, Kalman filter, Bayes estimation, neural networks, Hidden Markov Model [114].

Decision fusion deals with low coupling modalities and is close to cognition fusion. Specifically, it can be categorised as a task-oriented, hierarchical, probability-based, agent-based, or component-based fusion model. We consider our context fusion as belonging to feature fusion category using sequential audio and visual input (see Section 4.4).

2.4.2 Sketch and Speech

A sketch is an intuitive interaction used for communicating visual information between humans. A sketch, given its iconic nature, does not contain all the information necessary to express the concept completely. In particular, a sketch used to represent an object, in general, determines a category, and only a detailed version of it can produce the specific visual query. As in human activities, a sketch can be accompanied by speech, that tends to complete the information or insist on a particular concept, or even correct the information. Combining sketch and speech has been studied in order to accomplish different tasks. For example, Bischel et al. [16] describe a multimodal interface for interpreting the description of mechanical tools. They define a neural network with two layers that were fed with sketch features and speech features. The sketch features are defined as the set of geometric features extracted from the individual strokes and considering spatial and temporal relationships. The speech features are defined considering the temporal correlations between the stroke and the words. Adler et al. [115] specifically study the temporal correlation between sketching and speaking, generating a rule set

for segmenting and aligning the signals of the modalities. Therefore they used the aligned outcome as a source of interpretation. Adler et al. [116] develop a digital whiteboard able to understand both sketch and speech in order to create a collaborative environment. They produced a user study in which they limit the users, letting them use only command-based speech, annotation instead of drawing, unidirectional communication, and fixed set of graphical symbols establishing a vocabulary. Laput et al. [117] introduce PixelTone that is a multimodal interface for photo editing, interpreting speech commands for executing some specific editing process to the image while sketching is used for localising the area of the picture that needs to be changed.

2.4.3 Speech in Virtual Environments

Natural language interfaces have been applied extensively in many different domains such as databases [118, 119, 120], mobile devices [121, 122], car interfaces [123, 124], and home media systems [125, 126, 127]. Virtual reality might benefit from speech interaction as it could increase the sense of embodiment, and intuitiveness of the interface. Virtual reality gives a direct way to manipulate a 3D world. Implementing a multimodal interface requires some different issues to be tackled [128]: speech recognition, language understanding, and interaction metaphor. McGlashan et al. [129] describe a prototype system based on agents with simple dialogue capabilities, that the user can control with speech. In our study, we ask the user to describe the chair to have more elements to use during the search (see Section 4.6). McGlashan et al. [128] showed that spoken content retrieval through speech recognition potentially eliminates the need for text descriptions. The spoken content is translated by cascading adaptive speech recognition (ASR), and the resulting text is fed to a search engine to find the desired outcome. ASR becomes a critical component, and it risks being inadequate in real case scenarios, where a large dictionary and the language complexity can have an impact on its accuracy. Despite this, many applications were developed based on cascading ASR, such as SpeechFind [130], PodCastle [131], GAudi (short for Google Audio Indexing)[132], MIT Lecture Browser [133] and NTU Virtual Instructor [134]. In our study, we delegate the speech recognition and the language understanding to an operator that feeds an external software that elaborates the input and returns a set of predicted chairs (see Section 4.6). How, in general, a user communicates with the system is highly dependent on the task that is required and the framework we provide. Spoken queries to retrieve text content is a widely studied topic, and we designed an experiment to understand what is the best vocal query in term of the number of meaningful words that leads the user to better results. A detailed study of voice search topic [135] is out of the scope of our work.

2.4.4 Gesture and Speech in Virtual Environments

Human-centred interfaces improved user experience by exploiting a combination of modalities. Speech and sketch were used in different studies involving the virtual environment, as showed before. However, an interaction that engages both of them in a 3D model retrieval context received little attention. On the other hand, many studies use a combination of gestures and speech or handwriting and speech. Gesture, handwriting and sketching are three actions that convey visual information but they have important differences. Gesture drawing is a quick and generally shorted-time activity that aims to capture a fluid path. Consequently, the feedback to the user related to the type of movement is usually hidden or outlined and limited temporarily. In addition, the trigger that the gesture invokes is, in general, associated with a command. Hand-writing is the overlap between the visual depiction of a concept using the written language, so it is persistent information displayed to the user, and the system can interpret that. It is clear that the possibilities covered by handwriting can be exploited by speech interaction in a fast interaction loop. Sketching includes any drawing, requires visual feedback to the user, and in general, creates a context. It is not used as a trigger for a specific action. Also, while sketch and gesture can be represented with 3D coordinates, for handwriting, this is not possible, and a projection onto a plane is necessary before the interpretation phase.

Hauptmann et al. [136] applied the Wizard of Oz metaphor to study multimodal interaction for a 3D cube manipulation task. In that study, three conditions were defined: users with only gestures, with only speech and with the combination of speech and gestures. Project Quickset [104] introduced a framework where gesture and speech was firstly recognised in parallel, parsed and merged. This method permits the subjects to create content and locate them via vocal commands and gestures. Laviola et al. [137] implemented a system for interior design which made use of speech and sketch for creating and then manipulating virtual objects. Ciger et al. [138] introduced an application where the user interface includes a magic wand and spell casting by voice. Our research takes inspiration from Hauptmann's work and tries to demonstrate that the retrieve object model in 3D can be improved considerably by combining these two input channels correctly using a simple sequential model without the need of synchronisations or alignments (see Section 4.6.1).

2.5 Visual Metrics

We can classify image metrics in two categories: *quality* and *visibility* metrics. The first ones provide a unique quality score after taken in input the whole image. In general, they are trained and tested with mean opinion scores (MOS) [139, 140] with data achieved from user tests. On the other hand visibility

metrics [141, 142, 35] try to predict the visible differences between image pairs as the human visual system can do. They produce visibility maps (as shown in Figure 2.5), so local information on the image, where each value represents a probability of detection. Compared to the IQMs, they are more precise for tiny and slight artefacts even if they are not able to determine their gravity. Both the image metrics address different applications. In particular, visibility metrics are used by computer graphics applications that need to boost their performance without showing noticeable distortions.

2.5.1 Quality metrics

IQMs can be classified in *full reference* (FR) or *no reference* (NR). Full reference metrics represent the majority of IQMs. No reference metrics are more challenging to design as they do not rely on reference datasets, and simultaneously they do not provide an image database for metric assessment [143]. For these two reasons, we focus on full reference metrics. They compute visible local differences using both reference and distorted images. After a pooling process over the image, they provide a final quality score. Absolute difference (ABS) or Euclidean distance (ΔE) of the different channels (RGB) over the pixels between reference and distorted inputs are the most common techniques. Euclidean distance is used in Root Mean Square Error (RMSE) and in Peak Signal-to-Noise Ratio (PSNR) metrics. CIE ΔE 2000 (CIEDE2000) improves the prediction of visible differences by converting RGB values into a perceptually uniform colour space and using a colour difference equation.

Structural Similarity Index Metric (SSIM) [5] provides a global score calculating differences in local average contrast and intensities and moreover pixel correlation. SSIM uses a formula that considers three components (luminance, contrast and structure) as shown in Figure 2.6.

The Visual Saliency-Induced Index (VSI) evaluates similarly, but the local difference maps are based on four factors: two chrominance channels [144], the gradient magnitude, and the visual saliency. The Feature Similarity Index (FSIM) uses the gradient magnitude and, additionally, phase congruency calculated by the local difference map [145]. In the pooling stage, the weighting function is determined by VSI with the saliency map and FSIM with the phase congruency map. Despite VSI and FSIM

Figure 2.5: Visibility map (on the right) shows the probability of detecting the differences between a reference image (on the left) and a distorted image (in the middle).

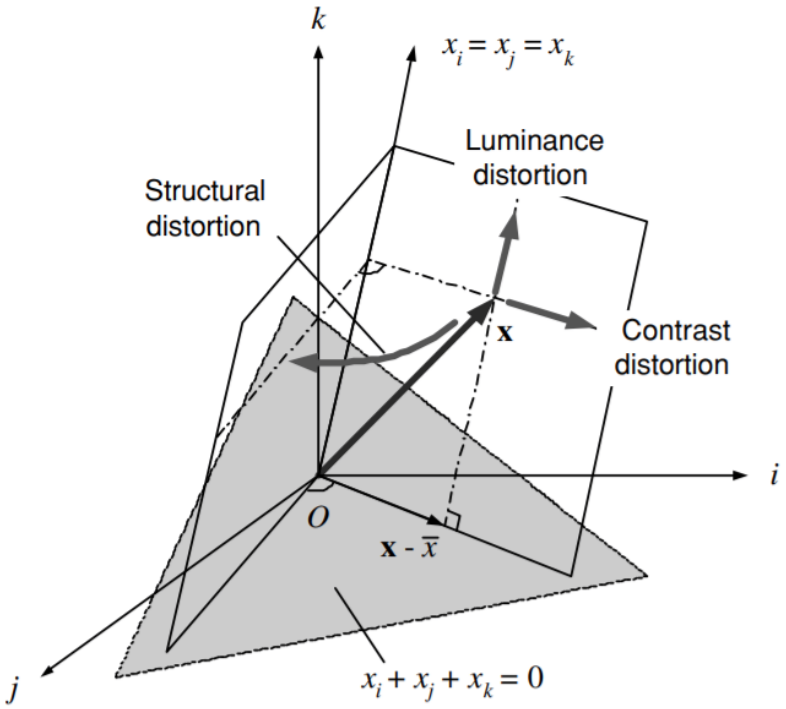


Figure 2.6: In the image space, SSIM separates luminance, contrast and structural distortion from a reference image (Figure from [5])

achieving difference maps at different points during the calculation, these outcomes' efficiency is not assessed as predictors.

Several overviews were completed about the IQMs [146, 147], that compare the results over image quality databases such as the popular collections LIVE [139] and TID2013 [140].

2.5.2 Visibility metrics

Visibility metrics try to predict the visibility of the distortions at each pixel. A large amount of data is required to train a data driven visibility metric. To overcome this requirement, these metrics are based on visual system models that limit the solution space by reducing the number of parameters involved in the training stage.

Spatial contrast sensitivity has been modelled by visibility metrics. sCIELab [148] used a spatial and chromatic contrast sensitivity to produces visibility maps. Although it improves the result of CIELab, this solution does not fit complicated images as it does not take into account contrast masking or contrast constancy. On the other hand, these kinds of problems are undertaken by more sophisticated metrics that consider luminance adaptation and frequency-selective visual channels [147]. These metrics are the Visible Differences Predictor (VDP) [149], Visual Discrimination Model (VDM) [150], and HDR-VDP [35]. Again, the accuracy achieved for complex images is not satisfying [151].

Butteraugli is a metric developed by Google within the project "Guetzli" [152]. It takes two

input images (reference and distorted) and produces different feature maps (edge detection map, low-frequency map) that are then used to create the final map with distortion prediction. This predictor is designed to detect artefacts coming from image compression, such as JPEG. It is not tested in different computer graphic tasks, where other metrics do not achieve relevant results [151].

A fundamental aspect of improving quality metrics via the training process is to collect a large number of images. This kind of data consists of manual annotations and needs to be performed on different types of distortions with supra-threshold, near-threshold, and sub-threshold magnitudes [147]. Alam et al. [153], for example, generates a local discrimination threshold for 30 images in 3000 patches. This does not represent the right method to collect a large number of images because of a boring task. Differently we opted for other options as described in [154, 151, 155].

2.5.3 CNN-based quality metrics

CNN-based methods represent a different approach to create a visibility predictor. In recent works [156, 157, 158, 159] some CNN-based IQMs have been introduced. These models are trained with a large amount of data, where small-size patches are created from reference and distorted inputs. The main issue for these patches is that they belong to the same image, so they inherit the same MOS value. For distortions that cover equally an image such as compression or noise, this hypothesis is reasonable. For other computer graphics artefacts that vary depending on the image position, this assumption can not be valid. In addition, even in the presence of compression artefacts, a unique MOS value is erroneous, for example, when contrast masking [147, 35] changes the probability of detection of a distortion. We overcome this issue by providing dense visibility maps generated during the annotation experiment.

In recent years, deep learning algorithms have been proved to improve performance in different areas considerably. Recent approaches try to improve IQM with SIFT and HOG [160, 161, 162, 163]. These features are the input of a regressor that predicts MOS. Improvements are shown when machine learning methods are applied to feature extraction and regression. A practical approach is to determine NR IQMs that do not need a reference image. Kang et al. [157] implemented a shallow CNN as a predictor considering the fact that low-level features are sufficient to determine the characteristics of a natural scene [5]. On the other hand, [164, 165] show how a deep neural architecture performs better than a shallow architecture. This advocates that higher-level features may have an important role in the results of IQM. FR-IQMs [164] implement an analogous architecture of NR-IQMs where there is a duplication of the basic NR neural network used for both reference and distorted images. The resulting feature vector is then passed to two fully connected layers for the final evaluation of the MOS value.

In the regression module, an additional branch is implemented to weight the patches. This

adjustment takes into account the spatial variance in the image. Moreover, it tries to compensate for the error introduced by considering the MOS value equal for all the patches produced by a specific image. In our work, we take inspiration from data-driven metrics to create a CNN-based visibility metric for predicting visible distortions. In addition, we collect an extensive set of images for the training phase that covers the majority of the typical distortion in computer graphics (see Chapter 5).

2.6 Gaps in Literature Addressed by this Thesis

Within the literature, there are several possible gaps and improvements that we have identified in the realm of 3D sketch in HCI and visibility metrics in visual perception. In this thesis we tackle:

3D Object Retrieval via 3D Sketch in VR

Sketch-based algorithms for 3D model retrieval received much attention in the last decade. Some approaches explored the generation of view-based descriptors [2], and they gather a collection of views and sketches that are used to train a neural network [70, 71]. On the contrary, methods based on volume-based descriptor suffers from the lack of resolution and are not suitable for searching fine details among a large collection of objects belonging to the same category. Transposing a sketch in an immersive environment is an additional challenge, and there is no relevant study on comparing 2D sketch and 3D sketch in that context. In addition, we noted that 3D sketch for retrieval purposes in VR is a novel study. We propose a system that helps the user to navigate a database using visual features and find the target model exploiting the combination between sketch and visual details of the objects.

Multi-Modal Interaction Based on 3D Sketch and Speech

The combination of speech and different input channels has a distant beginning [13]. Many approaches of later works use speech as a source of vocal commands more than using it as a way to describe an object. As the language in such contexts gives better results as a descriptive tool [112], we consider logical to juxtapose verbal description with the 3D sketch interface mentioned before. Visual features and features coming from a verbal description belong to two different spaces, although they try to depict the same object. We noted that such a VR system where immersive sketch and descriptive speech are fused to manipulate descriptors is a novel argument.

Efficient Visibility Metric based on Deep Neural Network

Determining the differences between two images is a widespread game. However, in computer

graphics, how the user perceives the output of the visual pipeline is an essential mechanism. This process induces feedback to produce a more realistic or distortion-free visual output. Visibility metrics are the entities in charge of determining the presence of artefacts in images, and in general, the results are modest. They lack generality [5, 145] and not always they provide distortion locality. Recently some deep learning approaches were proposed [164] in the context of IQM, evaluating a score for the images. We identified a gap within the literature and proposed a visibility metric based on a deep neural network trained with a large collection of distorted images coming from the computer graphics domain.

2.7 Summary

In this section, we analysed the most relevant studies in 3D sketch and visibility metrics. Our analysis included sketch-based image retrieval and speech interaction in the field of HCI. For the visual perception, we described FR visual metrics. Therefore, we detailed the challenges that this thesis addressed to fill the gaps in the literature.

Chapter 3

3D Sketching for Interactive Model Retrieval in Virtual Reality

2D sketching for retrieval tasks has been of particular interest for many years. On the other hand, 3D sketching has only recently been studied, thanks to the push of new cheap virtual reality and motion sensor products, and the improvement of pattern recognition algorithms. In this chapter, we try to analyse how sketch interaction performs in an immersive environment. Firstly, we want to understand the user response when both 3D sketch and several variants of 2D sketch interactions are possible methodologies for a searching task. Secondly, we extend this search mechanism by data fusion between sketch and model to improve accuracy and reduce the variance of the proposed results. The 3D sketch operates using an additional dimension comparing to the 2D version. This possibility means more expressive power because this extra-dimension can be perceived and exploited easily during the act of sketching. In real life, we deal with 2D sketches, and it is natural that the 3D version of it is something novel and for which we are not trained. In addition, it is more difficult to fill a surface with 3D strokes and this could require more attention. We investigate these aspects of the interaction, and we propose a method to improve the accuracy of the search: sketch and 3D model superimposition. Superimposing the sketch with a model helps the user in determining the spatial features of the object and, at the same time, can include additional fine details. In this chapter two studies are described. The first study (Section 3.3) was published in [166] and the second study (Section 3.4) in [167, 168].

3.1 Introduction

With the advent of new VR and AR headsets, there has been a rapid growth of interest in 3D modelling. Although many models may contain meta-data such as keywords and/or other data fields that outline their appearance and structure, these may be poor or unsatisfactory to express the complexity of

specific layouts. Furthermore, in many existing databases, these keywords and fields are frequently incomplete, insufficient, generic, and even incorrect. Thus query-by-example methods have become a very active area of research. In query-by-example systems, the user provides example entity instances. For example, the popular music recognition service named Shazam [169] requests in input a music sample (a few seconds are enough) and returns the matching song and related information. In our case, they are in the form of a sketch of parts of the object or scene they wish to retrieve. A search system then retrieves a 3D model that best matches the prescribed elements from a database. Users can interact with such a system in various distinct ways to retrieve a relevant object: inserting semantic information through a keyboard, selecting elements or characteristics from a Graphic User Interface (GUI), or depicting features by sketching. With the query defined, this activates the system to identify the most appropriate model by an appropriate distance metric or optimisation.

In the case of the sketch, traditional 2D methods used to search for a 3D model in an extensive collection may be tedious as 3D models require multiple views to be depicted. Beyond this, the sketching action also requires an expert user with a particular set of skills, such as understanding perspective and occlusion. By using virtual reality, this experience can be improved because ambiguity between views is greatly reduced. Firstly, the user no longer has to imagine the projections from 3D to 2D, and secondly, there is no need to select the most effective projection to sketch. Text queries or straightforward navigation through data collections are more conventional and popular methods. Although they are simple, appropriate and intuitive interface tools are required to avoid the searching task becoming cumbersome.

Therefore, the sketch method includes an implicit advantage compared to other query types: it requires no additional information e. g., tag or text or media associated with the model. Thus, our methodology focuses on the visual features of the objects. We describe the model in terms of the appearance of the structure and colour. We additionally consider the benefit of using a base object for the user to draw on top of, avoiding the need to draw significant structural elements and instead focus on fine details, making the methods more applicable for extensive collections.

A key challenge in sketch-based retrieval is that annotations in the form of sketches are an approximated representation of the real object and may suffer from being a subjective depiction and being over-simplified. These abstract portrayals are challenging to description methods and therefore require particular consideration. For image retrieval, methods rely on enhancing lines through gradients, GF-HOG [42] and Tensor Structure [170] or using multidimensional indexing structure such as NB-Tree [171], with more recent approaches based on CNNs [172, 173]. In contrast to 3D, the use

of sketching for retrieval has been limited to 2D projections for matching [3]. Matching 3D models requires to normalise models to have the same orientation so that a set of consistent and well-orientated images can be rendered and compared to the sketch (see Section 2.2.2.2). This view-based method is adopted and implemented as it allows an interactive experience where users remain immersed in the virtual environment and get responses with a minor delay.

Up to the present, there are various tools to allow the user to sketch (e.g., Tiltbrush, or Quill) in VR, but these focus on the sketch itself as the final result. Other systems enable free-form manipulation of objects by simple affine manipulation through drag points [174]. Conversely, sketching within a virtual environment both in its 2D or 3D variants as a retrieval method has received little attention. Therefore, we define distinct user studies to compare sketch-based retrieval methods to determine which one is the most accurate and satisfying for a user. In addition, we confront an improved version of the most precise technique to naïve linear browsing to demonstrate that sketching is a practical and usable method of exploring model databases.

In this chapter, we explore systems where the user remains immersed in a virtual reality display. We provide a base example of the class of object to act as a reference for the user. The user can then make free-form coloured sketches on and around this base model. A CNN model analyses the sketch and retrieves a set of matching models from a collection. The user can then iterate by making further correctional sketches (e.g. adding new pieces or style details to the model) until they find an object that closely matches their intended model. This leverages the strengths of traditional approaches while embracing new interaction modalities uniquely available within a 3D virtual environment.

In this chapter, firstly, we analyse sketch as input interaction for retrieval, comparing different modalities of sketching in an immersed environment (sec 3.3). Secondly, after confronting different descriptors, we upgrade the most precise method in order to improve both user experience and accuracy in the task of retrieving a model from a database(sec 3.4).

3.2 Sketch Interaction Overview

As shown in Figure 3.1, the interaction loop provided by the VR software allows the user to search throughout a 3D models collection by generating a sequence of strokes followed by an input submission. The system processes the input and then proposes a set of possible chairs from which the user selects the best match. Therefore, the loop can start again with a different model in the scene. The input consists of a sketch with or without a 3D model. When the sketch alone is used, only the visual information coming from the sketch is considered. When the model is included, a combination of

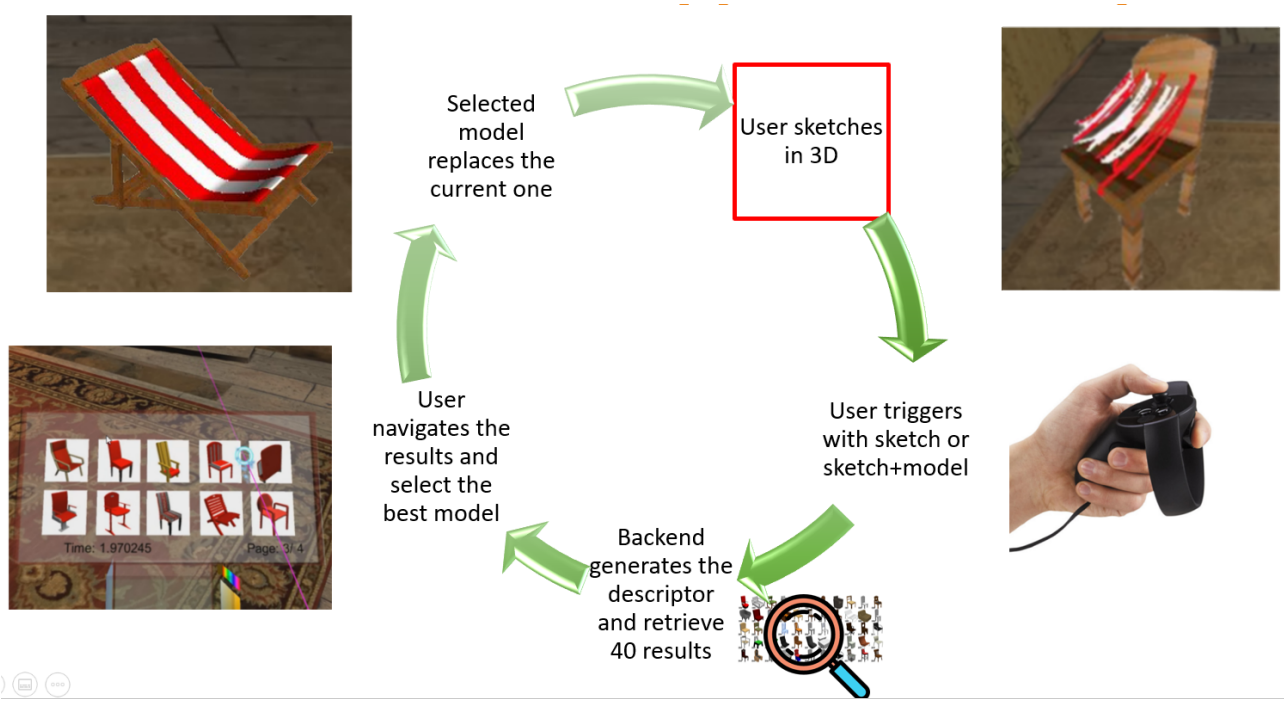


Figure 3.1: Interaction loop: the user sketch on top of the chair and send the sketch with or without the 3D model to the system. After receiving the proposed chairs from the system, he/she selects the best match.

sketch and model is sent to the system. The visual information of both the items is combined in a unique entity that takes into account occlusion and displacement between them.

3.3 Comparison of different Sketch Modalities in Virtual Reality

Sketching in VR allows users to design and create objects in 3D virtual space. To aid users in the creation of complex 3D designs, existing methods are often supported by a retrieval algorithm capable of finding complex designs based on a simple sketch made by the user by searching a model database. Common approaches can be divided into methods focusing on gestural interaction [82, 81] or techniques allowing to freely draw sketches [175]. Gestural interaction techniques are widely used to execute an action as a trigger mechanism or depict a simple trajectory in the design space. While gestures are generally easy to use, they are usually not suitable for characterising detailed features of an object. Gestures additionally limit the ability of the user to express their desires freely. This is especially true for the task of retrieval where flexibility is key to finding the relevant content, i.e., the so-called *needle in a hay stack* problem. However, both 2D and 3D sketches allow the user to convey complex structures, including their details. These techniques extend the scope of potential designs to a large number of objects within a collection with significant variations in terms of both shape, colour and texture.

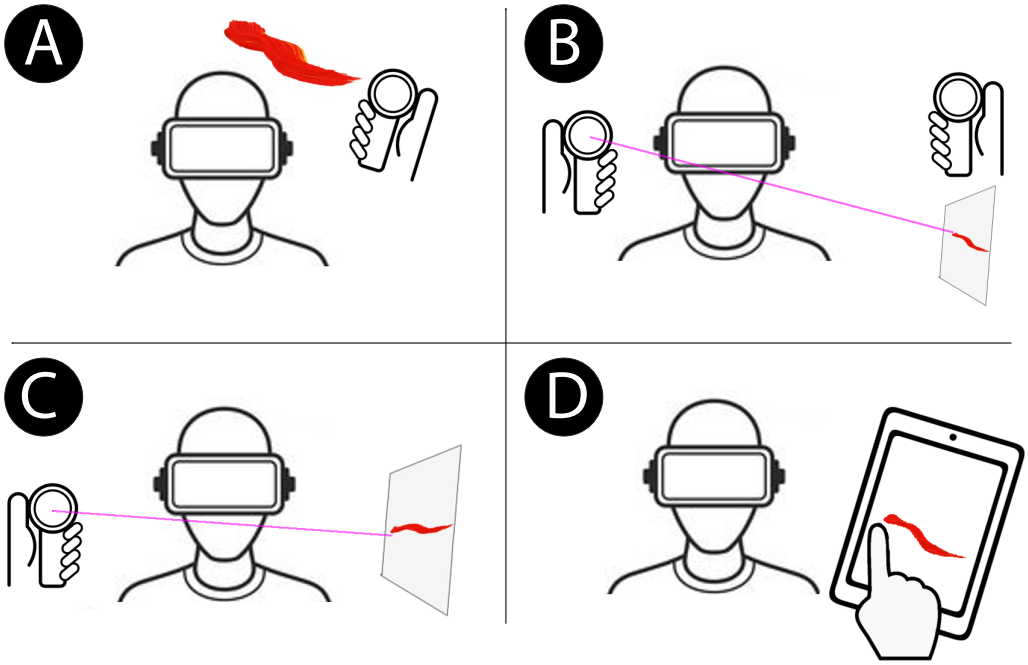


Figure 3.2: To understand supportive sketching within a virtual environment, we investigate sketching in virtual environments and consider 4 different interaction methods, i.e., (a) 3D mid-air sketching, (b) 2D sketching on a virtual tablet, (c) 2D sketching on a fixed virtual whiteboard, and (d) 2D sketching on a real tablet.

Despite the growing interest in methods for Sketch-based Retrieval [42, 176, 49, 177, 178], only few examples of such systems for VR have been proposed [80].

It is essential to understand how different interaction methodologies can impact both user performance and user experience. Therefore, we present a study to understand how users interact with physical and virtual devices framed in a retrieval context.

This work investigates different techniques (shown in Figure 3.2) for users to provide initial sketch designs as input for sketch-based retrieval algorithms in virtual environments. The contribution is two-fold:

- We compare four methods of Sketch-based Retrieval interaction in VR:
  - 3D Mid-Air Sketching, using mid-air drawing using a controller;
  - 2D Sketching on a VR Tablet, using a 2D tablet within the virtual environment;
  - 2D Sketching on a VR Whiteboard, using a VR plane to annotate the model;
  - 2D Sketching on a physical tablet, using a real world tablet tracked in VR to annotate the model with strokes.
- An extensive user study over the four methods identifying the advantages of methods with

regards to the user.

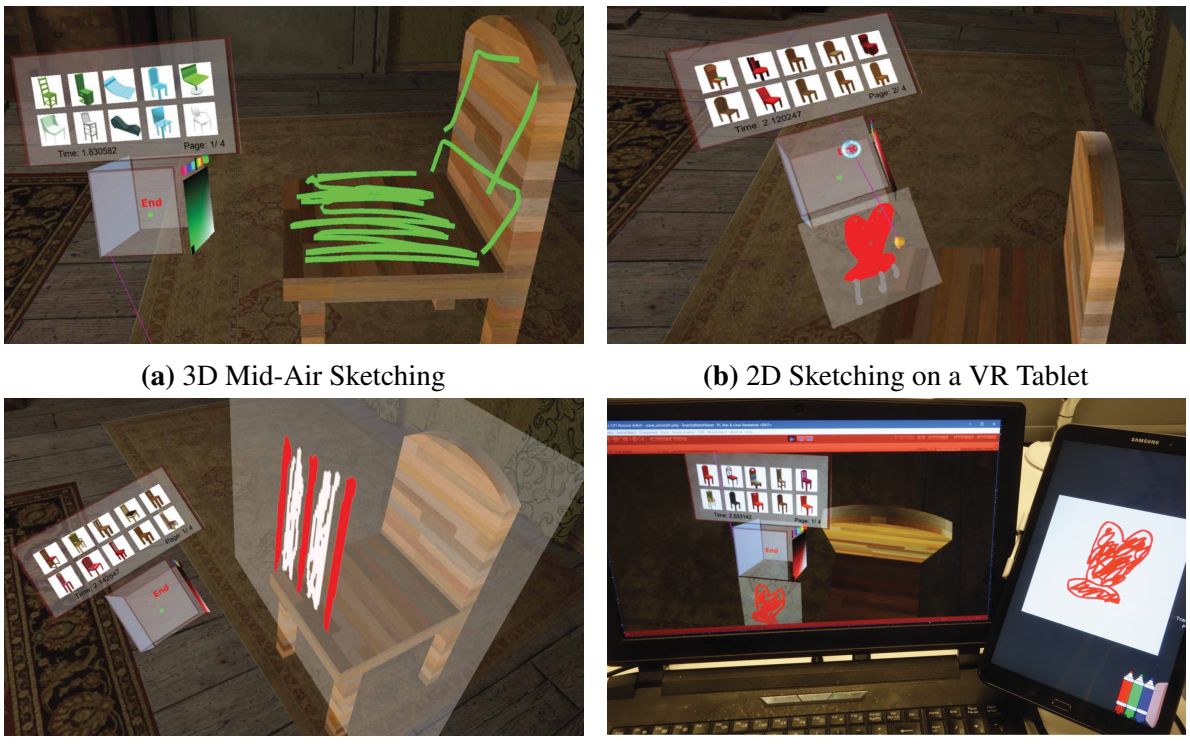
The details of this user study are described in Section 3.4.4, while the results evaluation in Section 3.3.4.

3.3.1 Sketch Modalities for VR

In the following section, we describe the implemented model retrieval system, with four sketch interaction methods and the back-end, which acts as a retrieval system. For each method, the user is immersed within a virtual environment and sketches either in 3D mid-air (3D Sketching), on a virtual tablet, a virtual whiteboard or on a tracked physical tablet.

3.3.1.1 Interaction Methodologies

We propose four distinct methods of interaction (as shown in Figure 3.3), three of them use 2D sketches generated on a different canvas and with different actions, and only one of them make use of 3D sketches. We selected the most natural approach for sketching in 3D, where the user traces the line in the 3D scene. Then for the 2D-based sketch, we use a virtual tablet to mimic the painter's palette, the whiteboard that gives a fixed and larger surface where to paint, and a real tablet to include physical feedback in the interaction. We outline them in detail below.



(a) 3D Mid-Air Sketching

(b) 2D Sketching on a VR Tablet

(c) 2D Sketching on a VR Whiteboard

(d) 2D Sketching on a Physical Tablet

Figure 3.3: Overview of the four implemented interaction modalities for sketch-based retrieval.

3D Mid-Air Sketching This method, shown in Figure 3.3a, is similar to existing systems for sketching in VR and is directly based on the method for 3D mid-air sketching([179, 69]). The user directly

sketches in 3D space using a hand-held controller. While holding down the trigger button, a virtual stroke is applied in the air at the current position. By dragging the controller through the air, strokes are extended in a continuous line. Once the trigger button is released, the active stroke is considered to be completed, and a new stroke can be initiated. There is no theoretical limit to the volume the sketch can occupy. The position of the sketch is arbitrary and can follow the edges of the chair model present in the scene or can be completely uncorrelated.

2D Sketching on a VR Tablet With this method, we mimic a natural method of sketching, but placed within VR. A 2D panel is attached to the user's non-dominant hand controller, see Figure 3.3b. As this method aims to simulate sketching on a portable tablet, the 2D panel was similar in size to a commonly used tablet device. The actual sketching of lines is done using the controller in the user's dominant hand. This makes the interaction technique a bi-handed approach as both hands are involved in the process of sketch creation, i.e., one hand performs the sketch while the other hand stabilises the drawing canvas. Here, the 2D sketch is not only limited in the third dimension, but also by the size of the panel.

2D Sketching on a VR Whiteboard Similar to VR Tablet, the whiteboard method provides a panel onto which the user can sketch in 2D, see Figure 3.3c. A familiar design paradigm, the whiteboard technique extends the size of the tablet to that of a larger whiteboard in order to provide more space for sketching. As the whiteboard is positioned on a fixed location inside the virtual environment, this method does only require the use of the user's dominant hand.

2D Sketching on a Physical Tablet Using a real-world tablet offers the user a physical surface to perform 2D sketching while immersed in the virtual environment, see Figure 3.3d. This mimics the most common technique used by digital artists.

The tablet is positioned on a table and requires a short registration procedure before starting a sketching session. While the tablet is still limited in drawing space, the physical feedback provided from the actual device aims to improve the stability during sketching. The user is able to sketch using her finger, thus this approach does not require the use of a controller for the sketching task. The user's hands are tracked using a LEAP motion device as the virtual environment needs to visualise the correct position of the hands of the user. This additional tracking is necessary as we noticed during a first implementation that the absence of the visual feedback for the finger position led to an unpleasant experience. This was mainly due to the user being unable to find the right location of contact between her finger and the tablet.

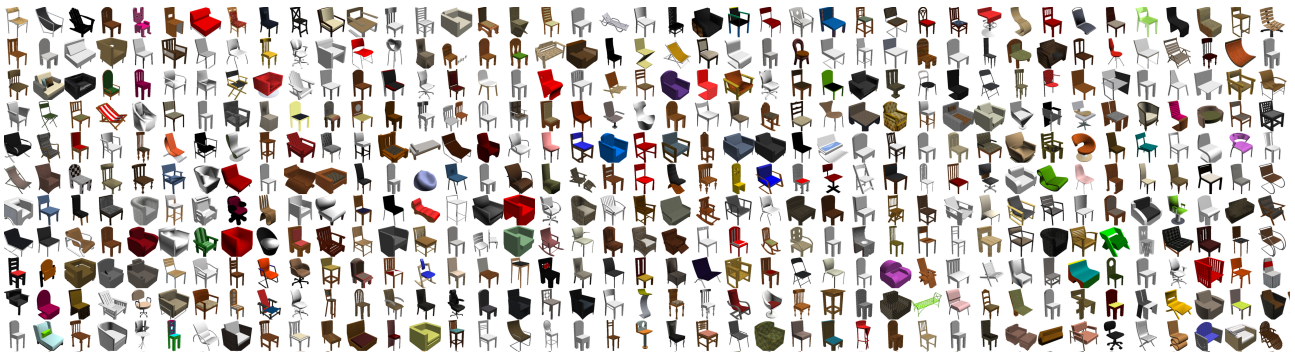


Figure 3.4: A random sample of ShapeNet chair subset.

3.3.2 3D Chair Collection

Sketch-based retrieval has had great success within 2D image retrieval, yet is still cumbersome when extended to 3D. This research proposes that by utilising recent advances in virtual reality and by providing a guided experience, a user will more easily be able to retrieve relevant items from a collection of objects. We explore the proposed methodology on ShapeNet [1]. ShapeNet is an extensive 3D model collection that includes a large set of model classes. We demonstrate our method to the subset of this collection that contains chairs, although our method is applicable to many classes of object. The chair subset is large and exhibits a large amount of variation that is particularly suitable for our method (see Figure 3.4).

3.3.2.1 What is a descriptor?

Having a computable representation of a 3D sketch is a challenging task. Multiple methods can describe sketch features. For example, a volumetric description subdivides the space of interest into voxels and fill the voxels where the sketch is present. This approach demands many resources, proportional to the cube of the linear resolution of a side of the cubic volume. Another approach is to take into account only some points of view of the sketch. We can generate the description using the images that originated from the view projections. This method requires fewer resources reducing the number of inputs. Besides, to improve the precision, increasing the number of points of view is not a difficult task and does not impact excessively on the system's performance. In a multi-view approach, the ways to generate the image representation, in jargon the feature or vector descriptor, are multiple. However, the outcome is precisely an array of thousands of numbers. In the second part of the chapter (3.4.1), we explore the different feature extractor algorithms and evaluate their accuracy.

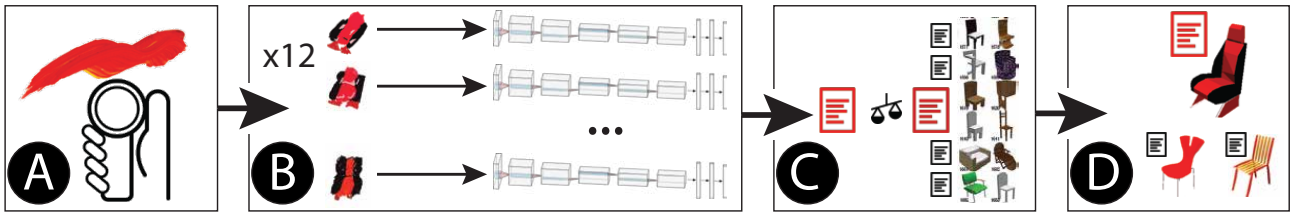


Figure 3.5: Overview of the system's model retrieval mechanic. Here, (A) the sketch created by the user results in 12 images (B) which are processed by 12 versions of the same CNN. After a max-pooling procedure, one descriptor is generated and (C) compared through Euclidean distance with the descriptors previously calculated for all the chairs of the collection. The search results are (D) a small subset of the most similar chairs from which the user can select.

3.3.2.2 Model Retrieval

To perform sketch retrieval, we implemented a backend that hosts a pre-loaded CNN model that generates a descriptor for the sketch image that is compared with the pre-computed database model descriptors. This backend answers the visual queries containing the sketches by producing a list of the 40 models that are considered most similar to the input sketch. We used the VGG-M Matlab implementation of Su [180]. This implementation provides a single visual descriptor after elaborating the snapshots taken by VR software. The method works by generating a set of structured camera views (snapshots) around the models. These images are then passed through the CNN, and a final descriptor is generated. During the generation of the snapshots, the cameras in the virtual environment move to different angles looking at the centre mass of the sketch and fits the viewport in the image. The CNN uses the VGG-M model [181]. An overview of the entire system is depicted in Figure 3.5.

3.3.3 Modalities comparison experiment

To compare our four interaction methods for sketching in VR, we designed and conducted a user study performed in our lab.

Participants: We used a within-subjects experimental design to help to reduce the number of participants and errors associated with individual differences. To counterbalance possible carryover effects, the methods were randomized between the users. As our independent variable, we distinguish the methods used to sketch, 3D sketch, 2D sketch on a virtual tablet, 2D sketch on a fixed virtual whiteboard and 2D sketch on a real tablet. We distinguish 3 dependent variables, namely the success rate, the completion time of the task and the number of submitted queries during the search.

A total of 5 participants (4 male, 1 female, $25-43$ age range with avg. 34) volunteered for our study. All participants had previous experiences with VR and already used an Oculus RIFT and Touch.

Apparatus: The rendering of the Virtual Environment was performed in Unity 2018.2.13 using an Oculus RIFT DK1 headset with a connected laptop computer. The specification of the laptop was:

Intel i7 CPU, 64 GB RAM with Nvidia GeForce GTX 980M graphics card. The interaction with the 3D environment was provided by both the Oculus Touch, i.e., two controllers paired with the headset, and hand-tracking using a LEAP Motion device. For the real tablet session, we use a Galaxy A6 tablet, tracked within Unity application via an Oculus Touch controller attached on the top right corner with Oculus Rockband VR Touch Guitar Mount Attachment.

Implementation: The virtual reality software contains one 3D scene. The scene consisted of a furnished room, with the addition of a chair when the system was initialised. After having triggered the system, the user can select the proposed models from a floating panel in which can scroll pages of models and display ten models at time. The panel is attached to the left hand, and the selection is performed using right-hand controller. Ten models were chosen so as to provide a panel that was small enough not to occlude large parts of the environments, but large enough that features in the chair were easily legible inside the HMD. The 3D sketching mechanism was managed through the generation of coloured lines. Lines are implemented as narrow strips that expose their wider section to the current camera. Therefore, each virtual camera, used for multi-view generation, renders the larger section of the strip independent from the sketch path. The user can colour using a palette connected to the left-hand GUI. The user can draw 3D lines in the virtual environment on top of the current model and can submit to the system using the controller's triggers. We also provided a simple UNDO function that acted on the sketch stack. We did not provide additional tools in order to force users to explore pure sketch interaction. The 2D sketching mechanism is achieved by drawing the sketch on a panel and inherit all the additional functions (palette, UNDO, etc.) developed for the 3D sketch. The back-end is a separate service thread in which a CNN Model is pre-loaded and ready to respond to user queries. This is triggered to produce a unique visual descriptor given the snapshots generated by VR application.

To maintain a reasonable computation time, the first convolutional layers (see figure 3.11) use stride 2, while the latter layers are used as normal. On average, the CNN process takes approximately 0.5 seconds to produce a descriptor after receiving input.

Procedure: Before starting the experiment, each participant was instructed on the searching task. A period of 10 minutes was dedicated to training the user to develop confidence with the controllers and the virtual environment. Between each method, users had 3 minutes of rest and they can perform the task seated or standing up. Upon completion of the introduction, the experiment commenced.

For each method, participants were asked to perform sketch searches for a given set of 8 different chairs. For each session, the participant started with a randomly selected sketch interaction method

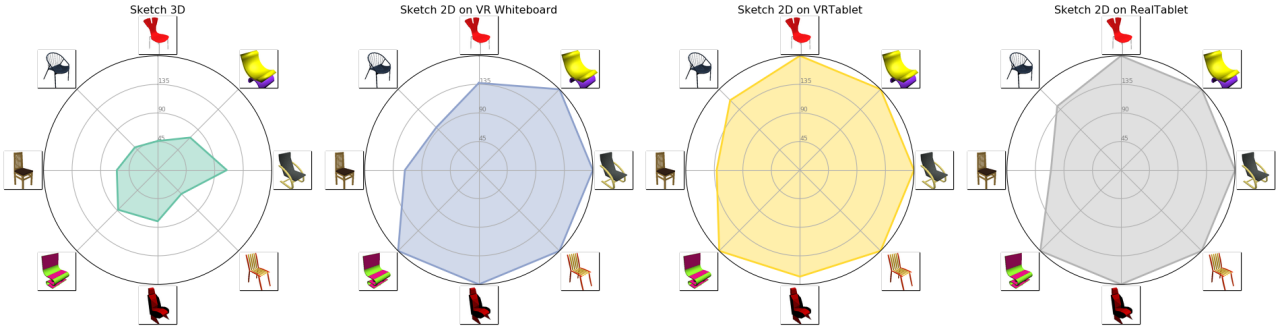


Figure 3.6: The inner circles in each radar represent 45 seconds. The centre of each circle corresponds to time 0. Each radar shows the average time to complete the task for each chair considering all the methods. The time is normalised to 3 minutes as the upper limit allowed for a search attempt.

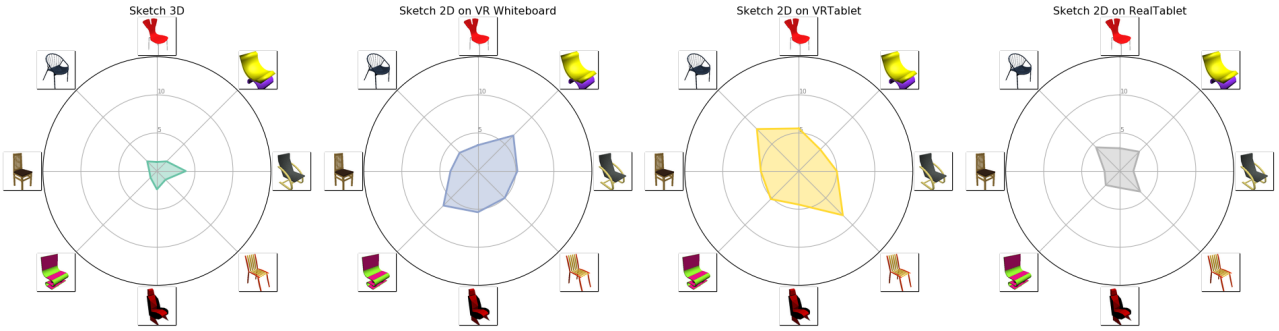


Figure 3.7: Number of search iterations for the different types of chairs for the different methods of interaction.

and performed the search for each target chair of the 8 proposed randomly. Using the selected method, the participant started sketching to initiate the search for the presented target chair. Upon confirmation, the system provided the user with a set of potential chairs considered to be most similar to the created sketch. The participant could refine the search results by editing or detailing the sketch. When the participant was satisfied with the results, the user terminates the test by clicking a GUI button. This selection concluded the search task and would replace the presented 3D model inside the scene. Each session is given a time limit of 3 minutes, after which the search was considered terminated without a successful result. The experimenter recorded the success rate and completion time for each task. The user's experience with the current sketching method was measured asking the user to rate the interaction on a scale from 1 (very bad) to 5 (very good).

3.3.4 Evaluating the Accuracy

We consider methods and models as independent variables and time to complete as a dependent variable. Because the two-way measure test does not follow the assumption of the normal distribution for the dependent variable, we run a Friedman test (non-parametric test) over our data considering methods as an independent factor. In addition, pairwise comparisons were performed (SPSS Statistics,

2019) with a Bonferroni correction for multiple comparisons. This test shows that the difference in the median for methods distributions is significant, with p-value less than 0.05. In particular, the pairwise comparisons show that 3D sketch method has the null hypothesis rejected with all the other methods. The Friedman test for the models variable is significant, with p-value less than 0.05. This result is caused only by one pair of models (over 28 pairwise comparisons). Model 8 has a distribution statistically different from model 3 distribution. We can explain this behaviour for model 8 because of the peculiarity of the chair. This chair has simpler features to identify and to draw, and less similar chair is present in the database. In this case, despite the performances of 2D methods, it happens that model 8 can be found faster than average and with the higher variance. On the other side, model 3 is the chair with the most common features in the data collection, difficult to find (worst mean value) and with one of the lowest variances.

We investigated the differences between the four different methods of interaction using sketches within a virtual environment. We evaluated our study over the 8 distinct chairs presented to each participant and 4 methods to test in terms of the accuracy of the returned model, time, and the number of queries to complete the task. To evaluate the accuracy we counted the number of successful searches among the total number of searches. The number of successful task completions for the 3D sketch was 37 out of 40 (92.5%), the 2D sketch with the whiteboard was 9 out of 40 (22.5%), the 2D sketch with virtual tablet was 6 out of 40 (15%) and the the 2D sketch with real tablet was 5 out of 40 (12.5%). To evaluate the efficiency, we measured the time elapsed from the beginning to the end of the search and count for each search the number of attempts to submit the sketch to the system. The average time among all the chairs for the 3D sketch was 71 seconds, the 2D sketch with the whiteboard was 156 seconds, the 2D sketch with virtual tablet was 169 seconds and the 2D sketch with real tablet was 166 seconds. The average number of attempts among all the chairs for the 3D sketch was 1,85, the 2D sketch with the whiteboard was 4.875, the 2D sketch with virtual tablet was 5.65 and the 2D sketch with real tablet was 2.9. We demonstrated how the variation in chairs effects the different methods in the radar plot of Figure 3.6. We showed in Figure 3.8b the cumulative number of attempts over the different methods of interaction. It can be seen that the 2D virtual tablet required a large number of search triggers, while the 3D sketch required the least. In Figure 3.7, we show how the difference in 3D model affects the number of required search triggers. These results mimic those of Figure 3.8b, but also demonstrate how different chairs provide challenges to the different interaction methods. Interestingly, for 3D Sketching and Physical tablet, the figure has a similar profile across models as opposed to VR Tablet and Virtual Whiteboard.

Finally, after the experiment, the user evaluated the different methods of interaction. Figure 3.8a shows the cumulative score. 3D sketch appears to provide the best user experience. However, the trend is inconsistent with the number of triggers of Figure 3.8b and Figure 3.7.

3.3.5 3D Sketch VS 2D Sketch experiences

The difficulty in finding some 3D models with 2D techniques in the database becomes evident from Figure 3.6. Despite all the methods being intuitive, the 3D sketch is more accurate and satisfying to the user experience.

In the case of 3D sketch the user can depict the target chair using more naturally the depth information, while in all the other methods user can draw only a 2D projection of a three-dimensional data on a texture. The real tablet method introduces physical feedback for the user as the drawings are generated by the finger when touching the surface, but essentially the output of the interaction is the same for all the 2D techniques. As each user performs 32 searches in total, they develop a plan to optimize the search, exploring how the input impacts on the neural network and exploiting eventual winning strategies for each method. Whereas 3D sketch interaction generates 12 input images from different points of view, the 2D sketch can contribute only with one. With 2D sketch, each user quickly developed the idea of selecting the most significant view angle and tries to depict that projection. Despite this, as the success rate for 2D methods is lower than 3D sketching. Some users tried to draw the different points of view in the same texture (top 'Whiteboard' Figure 3.9), in some cases achieving the target chair successfully. Also, while the 3D sketch does not require an accurate depiction, we noticed that for each 2D methods the user tended to detail the drawing to increase the probability of finding the target.

As a consequence, a typical behaviour noticed on the 2D canvas is that the user preferred to fill areas between the edges accurately. This disposition is not necessary for 3D where the system can interpret a few quick strokes as a filled surface. The motivation for this differences is again the lack of information of 2D sketch compared with the 3D counterpart, so the need to detail the 2D drawing as much as possible emerges naturally.

In terms of user interaction, for the aforementioned reasons 2D sketch modalities required from the user more query submission to the system (as shown by Figure 3.8b), more attention, a firmer hand, and in particular with virtual tablet two hands working at the same time. This could have caused discomfort after a few minutes of sketching and eventually, a loss of accuracy. Between 2D methods, the whiteboard shows better results than the virtual tablet for two main reasons: firstly because of the fixed texture to draw on, and secondly because of the larger canvas that can include more details.

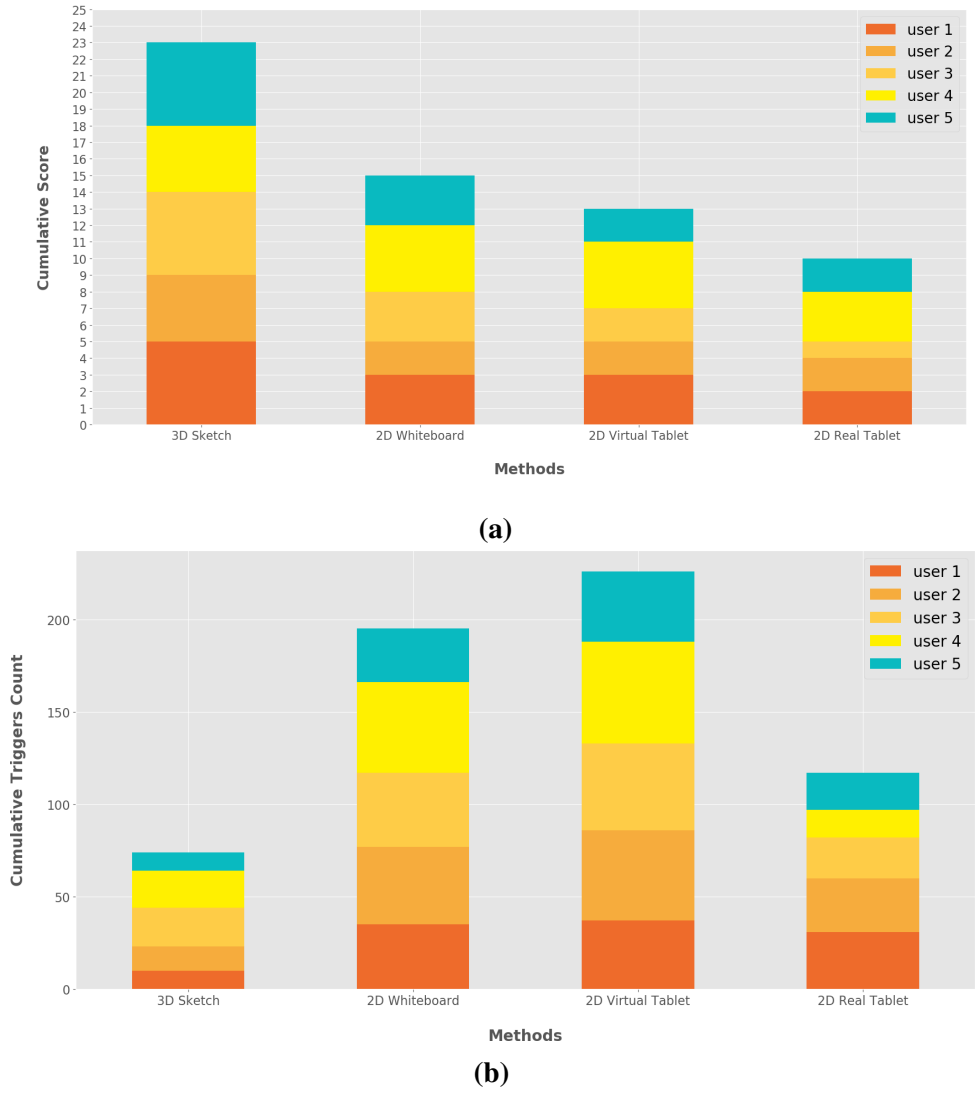


Figure 3.8: (a) Each bar is the cumulative score given by each user for a specific method. (b) Each bar is the cumulative number of search trigger events given by each user for a specific method.

Although the distance between the surface and the drawing hand is the largest between the 2D methods, this result shows that this aspect does not have a negative impact on the task.

The real tablet is interesting because of the physical interaction that is completely absent in all the other methods. While at the beginning of the session, this is a pleasant novelty, contrary to what was imagined, the sketch done directly with the finger was less precise if compared to the sketch generated by a remote controller. Moreover, mixing virtual reality with a tracked canvas was not sufficient to guarantee a decent user experience. The absence of the finger's positional information lets the user, still immersed in the scene, become disoriented during the drawing process as it is unclear where the finger is with respect to the canvas position. We solved this issue using a Leap Motion to track continuously the finger displaying an avatar, however, sometimes noise was introduced, and the user experience deteriorated rapidly. In addition, as it inherited the lack of performance of the 2D techniques, this

Figure 3.9: This figure shows in the first four columns some representative images from the 3D sketch. The fifth column is the sketch from the virtual tablet method. The sixth column is the outcome of the whiteboard method and the seventh column, the real tablet sketch. The last column is the image of the target chair.

method does not improve the results of the other 2D methods. This could indicate that even synthetic haptic feedback could aid in the drawing for the other methods, even in 3D sketching.

We observe a low number of triggering events both for 3D sketch and for real tablet techniques. 3D sketch had a high rate of success and a low number of queries that indicates the high efficiency of the method. On the contrary, even if the real tablet had a few triggering events, it does not mean that it is efficient. Its low success rate and problematic interaction lead the user to give up quickly, avoiding an extensive search as in the other 2D methods. Figure 3.8a shows that user experience is affected by the success rate, resulting in a better experience when the system is more accurate. With this study we confirm hypothesis [H1.1] that states that "3D sketch is an effective interaction to depict information in an immersive environment with the aim to retrieve models from a large collection."

3.3.6 Limitations and Additional Comparisons

Despite 3D sketch showing positive feedback from the user study compared to the other methods, we investigate some aspects and limitations of this system to improve the search accuracy or experience. Below we outline the most important ones:

Query Descriptors In our study, we use VGG-M deep descriptor. Recently, a large number of CNN architectures have been designed. We implemented our system as distinct modules that can be replaced easily. We aim to test different neural networks and machine learning models in order to do a comprehensive survey. These solutions could also be fine-tuned to the VR or learnt through active learning to become bespoke to the users' style.

Expanded Comparisons Our study compares four methods for generating sketches (3D or 2D). We could extend the study to other modalities, both 2D or 3D. One additional method could be a tracked pen coupled with the tablet to increase accuracy. An interesting follow-up would be comparing the user experience and accuracy obtained in a virtual environment with a real counterpart. In such case, the interaction with the physical tablet would be direct and more pleasant.

In addition, we aim to compare the sketch mechanism with more advanced user interfaces paired with state-of-the-art mechanisms for searching objects in immersive environments such as text input and faceted search. Moreover, we want to explore possible ways to integrate sketch with additional information coming from a more complex interface. Furthermore, several benefits can be achieved, providing functionalities such as brush size, erase and transform operations that could mimic the basic functionalities of 2D photo editing software.

3.4 Analysis and Improvement of 3D Sketch Queries

In this section, we describe the techniques used to extract the features from the sketch to create the feature vector. We present an improved approach based on 3D interaction for searching model collections based on annotations on an example model. This example model represents the current best match within the dataset, and sketching on this model is used to retrieve a better match. A novel aspect of this method is that we allow users to make sketches directly on top of existing models. The users can express colour, textures and the shape of the desired object. In addition, we evaluate different descriptors through a preliminary study in order to select the most accurate one, discovering that CNN achieves the highest precision. Third, we perform a user study to demonstrate the advantages of a sketch-based retrieval system in contrast to naïve search. We show that users understand the purpose and practical use of a sketch-based retrieval system and that they are easily able to retrieve target objects from a large database. Finally, our system is the first of its type to work online in an immersive virtual environment. This model retrieval technique can be broadly applied in editing scenarios, and it enables the editor to remain immersed within the virtual environment during their editing session.

3.4.1 3D Sketch Descriptors

Sketching within a 3D environment has been explored through stroke analysis [182, 183, 184], but little work has been performed to describe the set of strokes in a compact representation, i.e. descriptor, such as in SBIR [43] or SBVR [185]. Therefore we explore state-of-the-art model descriptions approaches. We apply four traditional Bag of Words approaches: SIFT [40], Histogram of Gradients(HoG) [41], Gradient Field Histogram of Gradients (GF-HoG) [42] and ColorSIFT [186]. It is worth noting that only ColorSIFT descriptor incorporates a description of colour. In addition, we apply a multi-view CNN architecture to describe the content of the model.

Each method generates a unique descriptor of the chair. To generate a single vector description of a model, the chair is projected into 12 distinct views as shown in Figure 3.10. Each view is then described by an independent model. This exhibits an early fusion approach which we describe for both deep and shallow descriptor generation methods.

In the multi-view CNN architecture [180] the standard VGG-M network of [6] is applied. This model consists of five convolutional layers and three fully connected layers (depicted in Figure 3.11). As in [180] the model is trained on ImageNet then fine-tuned on the 2D images of the 3D Shapes dataset. For each view of the model, the convolutional layers of the VGG-M are applied where the resulting descriptors are aggregated by element-wise max pooling. The result of the max-pooling is then fed through a second part of the VGG-M network (i. e., fc layers) where the second fully

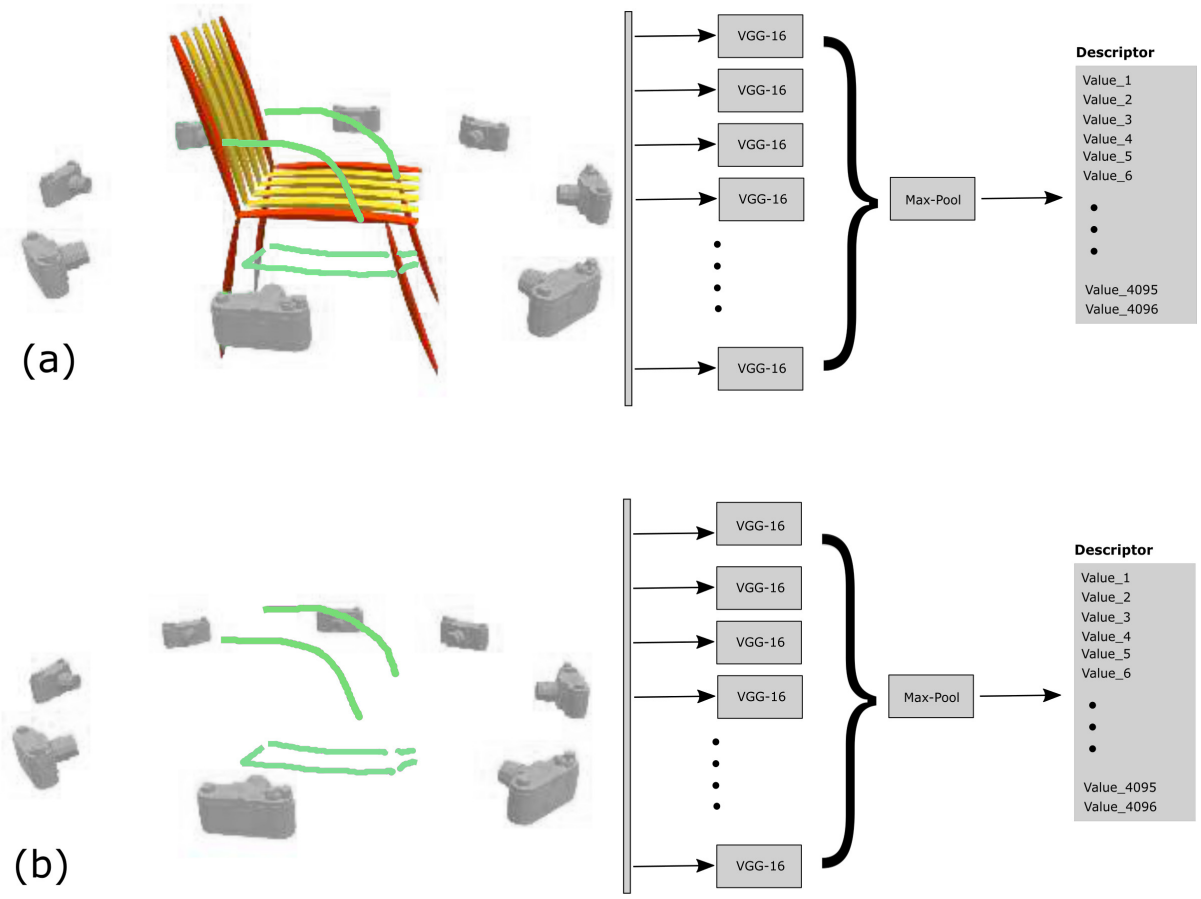


Figure 3.10: (a) CNN can be triggered with snapshots with both sketch and chair model. (b): CNN can be triggered with snapshots with only sketch present.

connected layer (fc_7) is used as the descriptor for the view (V) resulting in $V \in \mathbb{R}^{4096}$. The VGG-M network is trained once and shared amongst views.

For SIFT, ColorSIFT, HOG and GFHOG, we used the bag of words mechanism to generate a descriptor from all the views. The bag-of-words model was originally created in the context of NLP. When applied to a text, it creates the set of words present in that corpus, overlooking grammar but considering multiplicity. The distribution of these words is then used to characterise other text, and to create other classes for a classification. The same concept is used in computer vision where the words are replaced by features extracted with a specific algorithm. The BoW implementation is defined with $K = 1024$ clusters that represent the visual words and where the frequency histogram across views is accumulated to generate a singular descriptor, $V \in \mathbb{R}^{1024}$ for these methods.

We perform a preliminary evaluation of the descriptors for retrieval of models (See Section 3.4.3) and identify the approach of Su [180] to outperform the alternative methods significantly. Henceforth we discuss the approach in regards to this descriptor. An index is generated from the dataset by

repeating the aforementioned process over the dataset generating a matrix $M = D^{n \times 4096}$ where n is the number of items in the collection.

An alternative solution is to consider this retrieve task equal to a comparison task between images that represent the same object. In the case of a sketch that represents a 3D model, different algorithms coming from visual perception area could be applied to capture differences between images. We separately test the SSIM algorithm in the context of sketch generation. SSIM is a visual metric that accepts two images and produces a value between -1 and 1, where 1 means that the images are identical and -1 different. We calculate for each timestep of the sketch generation the SSIM result feeding the algorithm with the images coming from the target model and the snapshots of the sketch. The outcome is a time series of SSIM results. Differently from our belief that the similarity score increases, this time series does not show an increasing trend (ideally to 1). Structure similarity metric is widely used to evaluate perceptual image differences. It rates the degradation after processes such as image compression. Differently from other metrics such as PSNR (Peak Signal to Noise Ratio), it compares visible structures between images. On the other hand, sketch can be used to paint structures, but it is subjected to noise, user's ability, user's visual perception, that can alterate the similarity between a real object and the sketched one. Given these conditions, we discover that SSIM does not capture features coming from the sketches that try to depict an object, so we discard this additional comparison. Therefore, sketch transposition alters the shape and colours of a real object in such a way that SSIM is not able to evaluate the similarity of the structures correctly. We study in detail the visual perception algorithms in Chapter 5.4.

3.4.2 Sketch or Sketch/Model Online Queries

At query time, the multiple views are generated from the user's sketches and, optionally the current 3D model that is the best match (see below), and a forward pass through the network returns the descriptor. For simplicity and ease of comparison of results, we leave M to be linearly searched at query time. Improved efficiency could be achieved by using KD-Trees or other popular index structures. Therefore, we define the distance d as squared Euclidean:

$$d_i = |M_i - Q|^2 \tag{3.1}$$

where Q is the query descriptor. After comparing the descriptor with the descriptor collection, the system replies with the K-nearest models that fit the input sent. In our experiments, we use $K = 40$. The retrieved models are ordered by their respective r_i distance.

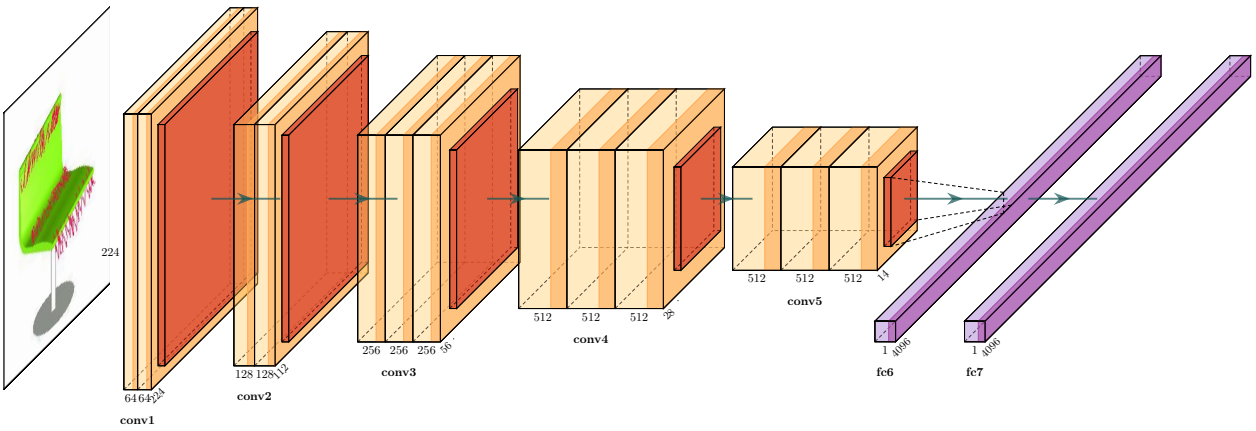


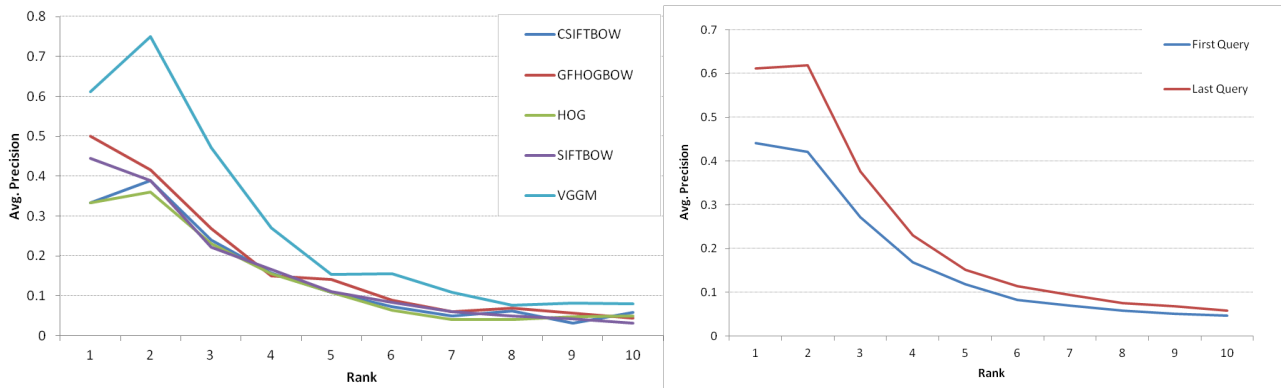
Figure 3.11: Each view is processed by the shown VGG-M architecture model [6]. As demonstrated in Figure 3.10 the network is split after convolutional layers the final Multi-View descriptor is the output of the network a vector of 4096 scalars.

We provide the user with two ways to perform the query: sketch-only query or both sketch and model query. This is achieved by enabling or disabling the visualisation of the model (see Figure 3.10). After the system proposes results if the user's target model is not present the user can edit the sketch or conversely can replace the current model with a new one that better matches represents the desired target. Such a possibility helps the user to minimise the time sketching: they can focus on sketching the missing or different parts relative to the current best match model. This facilitates a step by step refinement to navigate through the visual feature space of the collection, commonly achieving the target model only after a few iterations. In the current implementation (see Section 3.3.3) the response time after each user search request is 2 seconds. This is sufficiently quick to allow a tight interactive loop between sketching and querying. Users are free to either make a complex sketch that will likely match on the first attempt, or add features to the model in several iterations, thus facilitating a 'walk' through the model collection towards the desired target.

3.4.3 3D Sketch Descriptors Evaluation

We perform a preliminary study using a set of six queries over the different descriptors. We evaluate their retrieval precision with regards to a set of criteria for the returned model. Following the approach of Collomosse [54, 185] we evaluate the precision in terms of these different facets of the retrieval, therefore for each correctly returned facet of the model, the score is incremented. These correspond to: 1) Structure – the majority of the parts arms back, seat, legs; 2) Style – curvy, straight, with many lines; 3) Colour – dominant colour matches query. The evaluation was performed by a human.

This study aims to identify the descriptor that achieves the best precision for the search task. The



(a) Average precision calculated across ranked results from preliminary study.

(b) Comparison of the first and last query average precision from user study.

Figure 3.12: Average precision.

most accurate method is then used in the user test. In addition we prepared two sets of queries, the first are pure sketch queries, while the second are a combination of the sketch and the model. We considered the top 10 retrieved chairs proposed by each method, ranked from position 1 to position 10. Each rule can assign only one point if matched and focuses on a specific feature of the model. We formalised the rules as follows:

1. we consider four components of the chair: back, seat, arms and legs. Where if more than 75% are similar to the target, the result is considered correct;

2. if the proposed chair shows a dominant style (curvy, stripes, convex, etc.) similar to the target chair;

3. if the proposed chair shows a dominant colour similar to the target chair.

With each result receiving points for the facets, a final score in the range of $[0,3]$ is calculated, which is then normalised across facets and queries for a result in $[0,1]$. The precision is calculated from the scores for each result, using the equations:

$$P_r = \frac{\sum_{i=1}^{r} S_i}{r}, \tag{3.2}$$

where P_r is the average precision for the rank r, S_i is the score for rank i assigned by our metric. We compare SIFT, ColorSIFT, HOG, GF-HOG and VGG-M, calculating the average precision for each chair of the top 10 retrieved models. VGG-M method outperforms all the other methods using sketch and model queries (as shown in Figure 3.12a) and also using only sketches.

We calculate Mean Average Precision (MAP) for each descriptor. For the sketch and model

Figure 3.13: The scroll method provides a simple scrolling panel for navigating the database of all the chairs.

queries, VGG-M's MAP achieves 0.28, followed by GF-HOG with 0.18. This pattern is similarly reflected within the Sketch only queries, with VGG-M's MAP highest at 0.22, followed by SIFT with 0.13. Therefore, we perform the user test using descriptors generated by VGG-M. This analysis confirms hypothesis [H1.2] that states that "deep learning models are more effective for extracting features than well-known detectors and descriptors when applied to sketches multi view 2d-sketch supervision."

3.4.4 Query-By-3D Sketch VS Linear search in VR

We designed an experiment to compare two methods: the proposed sketch-based method, and a naïve scrolling panel method. For each session of the test, we first showed the participant the twelve views of a target chair as generated for the descriptor. We then asked the participant to retrieve the chair from the database, using one of the two methods. For both methods, the participant started in a scene of a furnished room where a chair is positioned on the floor to the user's left-hand side. We perform this initialisation step to minimise the required hand travel distance avoiding any mobility bias. We tracked the success rate, the time to complete the task and a subjective evaluation of the user experience through a questionnaire.

The scroll method consists of finding the target chair from the entire collection of 3370 chairs using a panel that shows ten chairs at once and which can be scrolled forward and backwards very quickly. After the user starts the session, the chairs are randomly shuffled to prevent the recall of the order from memory. The user then searches for the target chair (see Figure 3.13). When the user is confident that they have found the chair, they select it from the panel in order to replace the current chair in the room. When the participant clicks the end label, the time required to complete the task is

Figure 3.14: An example of a user's sketch within the sketch interface. The query is comprised of coloured 3D strokes drawn on top of a chair model.

taken, and the session is finished. For the sketch method, the user makes coloured sketches on top of the initial model (see Figure 3.14) and then uses the hand-held device to trigger the search method. The system proposes 40 chairs as the outcome, shown ten at a time in a scroll panel that is navigable in the same fashion as the scroll method. The participants would iteratively sketch, triggering the retrieval system or selecting models from the 40 suggestions, then continue to refine the sketch. The search refinement process continues until the target chair is located, and the user can terminate the session.

3.4.4.1 Experiment procedure

The apparatus is the same described in Section 3.3.3. The software implementation differs as we used only 3D sketch interaction, and we implemented a linear search through the database by panel navigation. All participants are asked to complete an introduction form with basic information related to their previous user experience in 3D software and VR applications. Each user performs two sessions of tests. Where each session is comprised of two sub-sessions. In each sub-session, the user performs three search tasks for different chairs models with one method, and then the same three searches with the other method.

Participants were instructed before each of the four sub-sessions with an application demo in which it will be shown the modality they had to use. In addition, they could select to practice for a short time to familiarise themselves with the interaction. Each of the search tasks was started by asking the user to look at a particular target chair with the instruction find it using the selected method. For

the sketch-based method, we instructed the user to use the style they prefer, which could be based predominantly on making a single sketch or on system interrogation with multiple iterations of model replacement. Each user was allowed to perform the task seated or standing. An upper time limit was defined as 4 minutes in order to keep each user session slightly less than one hour. In the event, the user was unable to locate the target chair within the time limit, or the wrong chair was selected, the search was considered a failure and the time cropped to 4 minutes. The two sessions differed in starting method used and from the different set of target chairs; thus, the order of the methods is counter-balanced, and each subject uses both methods twice. We split the users into two groups: the first group started with the naïve scroll method in the first session, while the second started with the sketch method. In total, each participant performed 12 searches. In this way, we were able to analyse the task completion time considering the contribution related to the different techniques, to the chair types and to the learning curve effect of VR interaction. We choose six different chairs with specific structure and colours. In particular, both striped and curvy shapes are present in the sets with a variety of different colours, as shown in Figure 3.15. After completing all four sub-sessions (12 search tasks), participants filled in a final form with their rating on user experience and level of confidence for both scroll method and sketch method. The scale of the rating was expressed in the form of a scalar from 1 to 5.

3.4.4.2 Improved 3D sketch user study evaluation

Our user study consists of 30 participants recruited from the university and general public. We split the participant into two equal size groups (15 users per group). The first group of participants started with scroll method, while the second group started with the sketch method. Twenty of the participants were male (10 female) while the average age of the participants is 26 years. Each of the participants in the study performed six scroll and six sketch tasks, giving a total of 360 search tasks across all participants. The tasks splits are demonstrated in Figure 3.15 with regard to group and session (see supplementary material for user final queries), i. e., twelve trials per user, with 15 participants doing the first task with scroll, 15 doing the first task with sketch.

The number of successful task completions for the scroll method was 119 out of 180 (66%) and for the sketch method 171 out of 180 (95%). In Figure 3.20 we show the total number of completions for each method for each task, in the order that participants completed the tasks in their respective groups. This graph shows the impact of individual tasks being found easier or harder by the participants. As there does not appear to be a trend over the sequence of tasks for the sketch method, it demonstrates minimal learning required and the intuitive nature of the method.

The task completion performance for the sketch method can be affected by the complexity of the target model, where difficult models are challenging to depict. The participant may have improved their depiction ability or efficiency with the system, but this can not be conclusively drawn from these results. While the significant factor for the linear search is the position within the dataset. It also can be seen a much larger variation in completions per task for scroll than sketch. For task three, only 3 participants completed the search with the scroll method. This in comparison with sketch, the minimum number of completions was 12.

We show the time to complete all tasks in Figure 3.18 for each of the methods. We can see that the distributions are very different, with a cross-over point at around 60 seconds. This can be explained by the fact that the page number largely determines the completion time for the scroll method that the result appears on, while for the sketching method there is an additional interaction overhead for completing the query sketch and the search time.

We compared the average time to complete all six models for the sketch or scroll methods in a paired-comparison per user. Each pair comprised the average time to complete all six sketch tasks and the average time to complete all six scroll tasks. Additionally, any failures to complete were clamped to 240s (4 minutes). The median time to complete the sketch tasks was 99.8s, and the median time to complete the scroll tasks was 156.5s. We used the exact sign test to compare the differences because of the distribution of times and the clamping on failure. This showed that the difference in medians was significant, with p less than 0.0005. We asked participants to report feedback on user experience. In Figure 3.19, we show an average rating of sketch and scroll methods for all users. We can see quite clearly that users strongly prefer the sketch method, with only two users rating the scroll method as favourite one, four showing no preference and the remainder (24) preferring the sketch method.

Qualitative examples are shown in Figure 3.16 and Figure 3.17, showing the types of sketch created by the participants. We discuss further the difference between the types of sketch in Section 6.

Finally, we reflect on the development of the precision of results across the session for users in the case more than one query was performed. Our purpose is to quantify the improvement between the first and the last query, without considering the cases in which the user found the target chair after the first interaction, and therefore considering the refinement of the results over time. We evaluate using the same mechanism as in the comparison of descriptors (Section 3.4.3) but solely for the selected descriptor VGG-M, in Figure 3.12 (b) we can observe for each rank an improvement of the scores achieved by the last query compared with the first. To quantify this improvement, we calculated the MAP for the first queries that achieves 0.17, while the MAP for the last queries is 0.24, showing

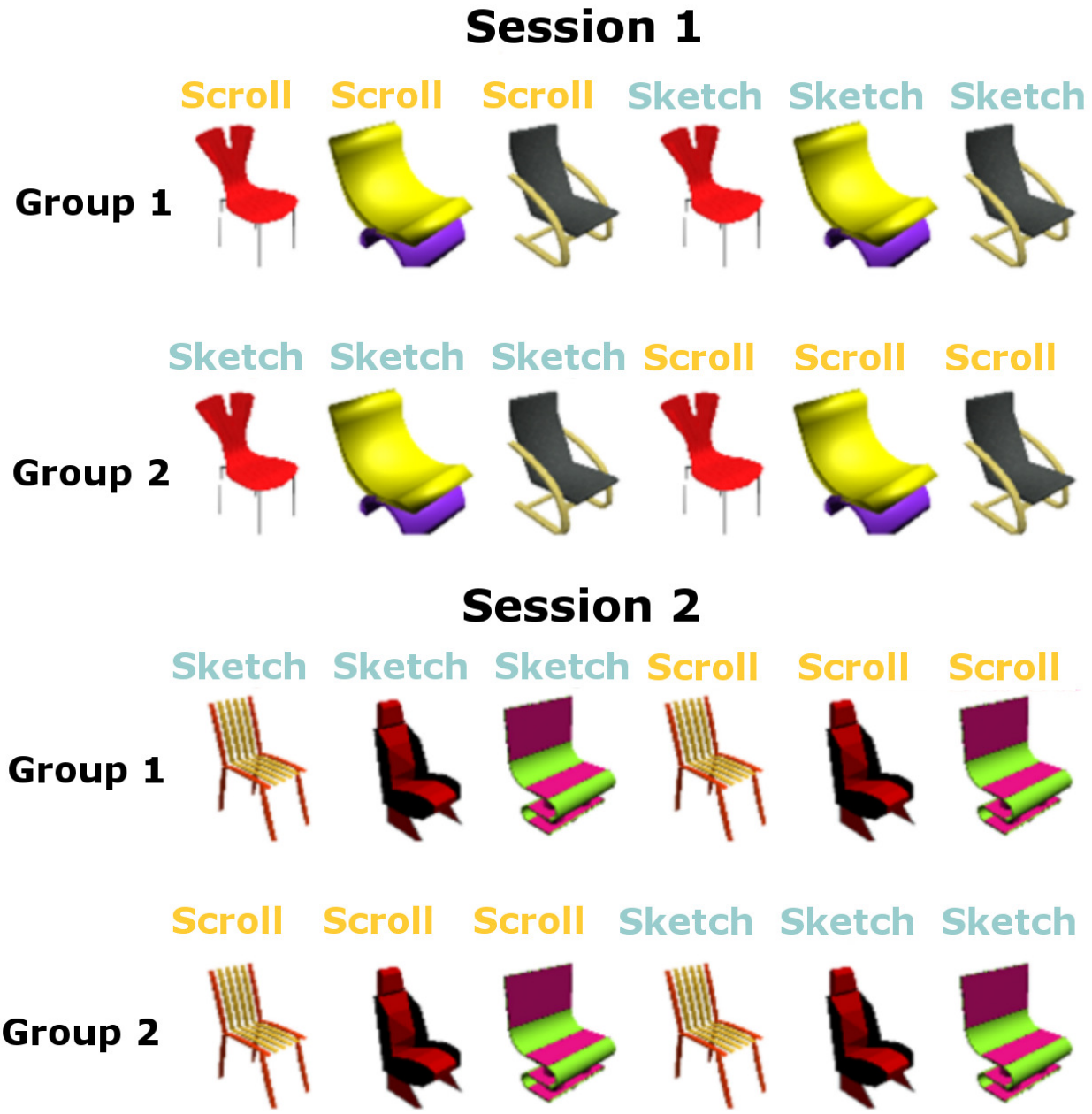


Figure 3.15: Two groups of 15 users are created. The first group performed the scroll method as the first method for the first set of chairs, then with the sketch method for the first set of chairs, then swapped the methods over for the second set of chairs. The second group did the opposite order of methods on the same order of sets of chairs.

Figure 3.16: Examples of users that successfully triggered the system using a combination of sketches and model. The left column contains the target chairs, while the other columns contain a subset of the snapshots used by the system.

Figure 3.17: Examples of users that successfully triggered the system using only sketches. The left column contains the target chairs, while the other columns contain a subset of the snapshots used by the system.

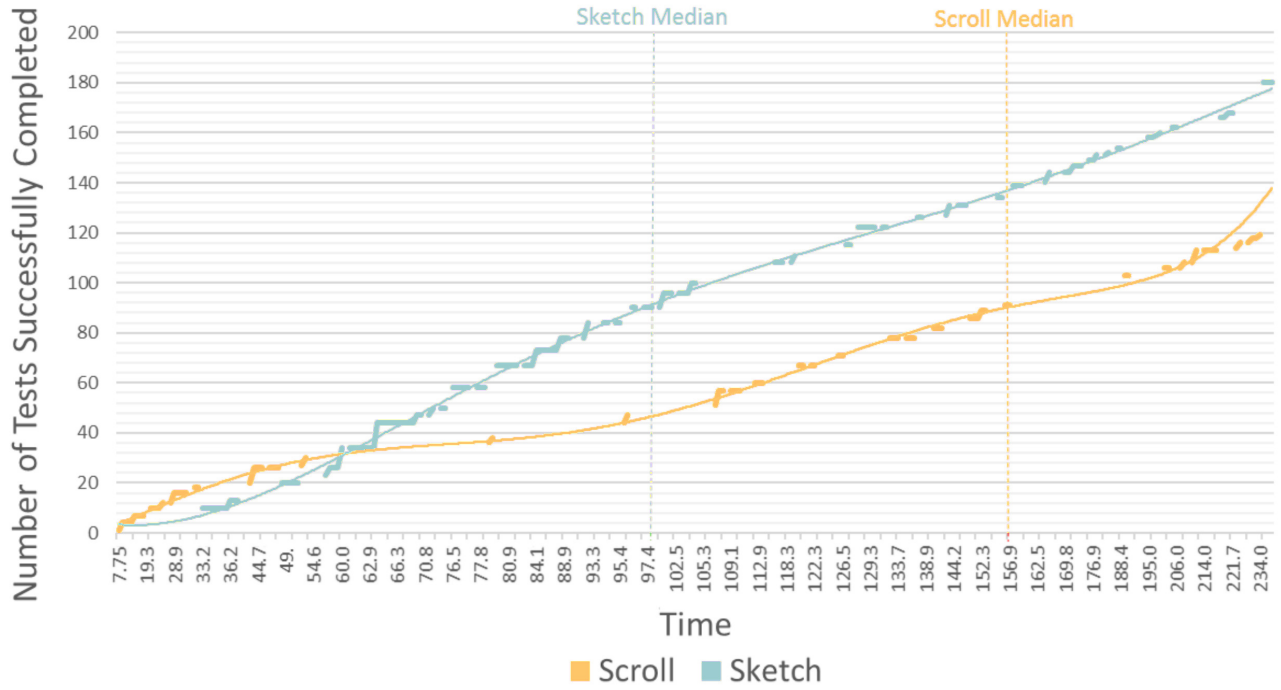


Figure 3.18: Cumulative time distribution for the scroll and sketch method. If the target chair was not found within the time limit (240 seconds) the time is limited to this.

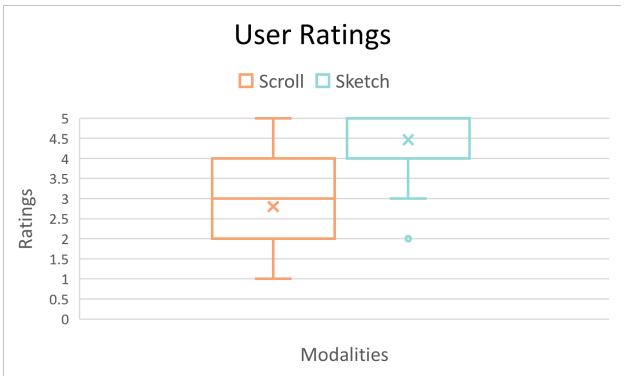


Figure 3.19: User ratings for scroll and sketch method are summarised in this box chart.

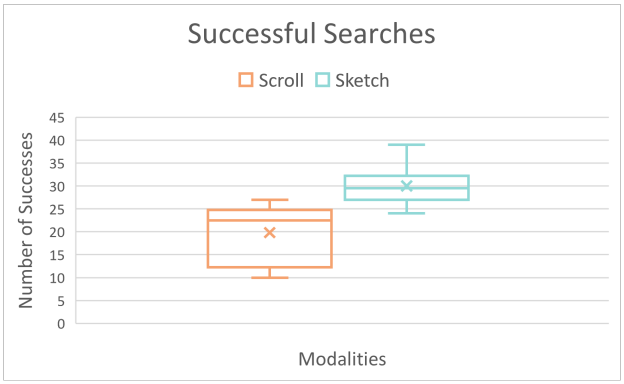


Figure 3.20: Box chart that shows the aggregate of successes for sketch and scroll methods for all the chair models.

improvement during time.

3.4.4.3 Comparison between 3D Sketch Descriptors

Our preliminary study compares the precision achieved by different descriptors in order to decide the most accurate method for the user test. We defined the metric rules in such a way that it avoids assigning additional points if the target chair is present in the results. Despite this, VGG-M clearly achieves the highest precision scores for all the top 10 ranks. Consequently, this result shows that VGG-M descriptor is the most accurate in retrieving different facets (colour, style and shape).

3.4.4.4 Improved 3D sketch evaluation

Our purpose is to explore 3D sketch interaction for object retrieval in order to understand its validity and possible developments. Therefore, we designed an experiment to identify different user approaches between our method and a simple linear search. In addition, we avoided to include complex functionalities during sketch phase to study the effectiveness of pure sketch interaction.

Our experiment shows that it is possible, through an iterative process of sketching and model selection, to perform an effective search for a model in a large database while immersed in a virtual environment. Further, the accuracy and completion time are significantly improved on naïve scroll method and the participants also prefer the sketch based approach.

While the scroll method represents a baseline with a clear and linear work-flow to the user, the sketch method allows different strategies. In general two different techniques emerged from the experiment: sketch only as shown by examples in Figure 3.16 and sketch with a model as shown in Figure 3.17. The first and more intuitive approach is to make a single sketch and detail it step by step until most features of the chairs are resolved without replacing the model. The user can interrogate the system to have feedback but essentially will continue to sketch. The downside is that the user can waste time on detailing a sketch and, in addition, can depict features that are not relevant. Determining whether features are relevant or not is not a trivial task for two reasons. The first one is that different users will over-rate the saliency of the feature (e. g., it may be an uncommon feature but the descriptor has not captured it). The second one is the possibility that the specific feature is common to many objects of the database. Both cases can lead to an unsatisfactory answer from the system as it proposes a chair set without that feature or conversely many chairs containing it.

The second approach is to only model differences to the current object: that is, the user queries the system and then only adds features that are different in the target object. The sketch is usually started again after each query. The advantage of this method is that the quick response from the system (˜2 seconds) enables fast iterative refinement. Every time the system receives a different combination of sketch and model, it will retrieve a different set of chairs. This method requires more experience from the user, but after a few iterations, we observed several participants starting to adopt it. In addition, we demonstrate, through the comparison between first and last query outcomes, that user improves the precision as the search progresses with time, increasing the similarity of the facet of the retrieved models with the facet of the target. Therefore we consider valid hypothesis [H1.3] that states that “feature descriptors generated by the combination of sketch and 3D model in the context of an interactive loop, are an improvement in terms of accuracy.”

3.4.5 Overcoming the Limitation of a Single Category

Despite the benefits of using sketch and the positive feedback from the user study, several aspects could be investigated to improve the search accuracy or experience. These are outlined below:

Multiple Object Categories In addition to working with chairs, we performed an additional experiment with the table collection within the ShapeNet database. We verified the same behaviour of the system using the proposed approach. As the approach has no fine-tune training for the chair object category, it is plausible that results can further be extrapolated over the larger collection, with an initial object category selection at initialisation.

Description of 3D Sketch The proposed method uses the same technique for describing the sketch and the model. Commonly within SBIR techniques deform the image to just a simple line (edge) representation for it to be more comparable with a sketch. Inspired by this, a technique to represent the object in a more simplified way. Or alternatively developing a method explicitly designed to handle 3D sketches would improve retrieval precision.

Accuracy of the search Our system compensates for the accuracy of the search that sometimes is not high by presenting a reasonable number of chairs in the result set. This strategy is reliable as 40 chairs are easy to navigate and show the user the new features appearing in the answer. On the contrary, a result set consisting of 5 chairs could reduce the accuracy of the search drastically since some features are not perceived correctly from the neural network and struggle to show themselves in the results. In particular, despite colour is detected by the feature extractor, it is difficult for the system to pair the colour with chair parts. ShapeNet contains a large number of different chair shapes and colours, and rarely it is possible to find colour permutations or different colours applied in chairs with the same shape. Therefore the issue mentioned above does not appear and is hidden. It becomes manifest, generating a collection with fewer shapes, but with a large number of chairs with permuted colour from a limited amount of possibilities. It will be the argument of study in the next chapter.

3.5 Conclusion

The benefits of virtual reality in the field of scene modelling have been investigated for several years. Previous research has focused on free-form modelling rather than developing a way to retrieve models from an extensive database. Current strategies for navigating an existing dataset use queries on tags or show the user the entire set of models. In addition, large collections can suffer from a lack of meta-information which hampers model search and thus excludes part of the dataset from query results. In this chapter, we described two studies. The first study (Section 3.3) aimed to compare different

sketch modalities for model retrieval in an immersive environment. We proposed a novel interaction paradigm that helps users to select a target item using an iterative sketch-based mechanism. We evaluate accuracy and user experience after conducting a within-subjects experiment. In the second study (Section 3.4) we focused on the 3D sketch in VR, and we improve the interaction with the possibility of combining sketches and a background model to form a query to search for a target model. We run a study to determine the most accurate descriptor. An experiment collected information about the time taken to complete the task and user experience rating. We compared our method with a naïve scrolling selection method. The sketch-based method was clearly preferred by users and led to a significant reduction in search time. We thus believe that sketch-based queries are a very promising complement to existing immersive sketching systems.

Chapter 4

Multimodal Approach Fusing Sketch and Speech in Virtual Environment

We pose the question of how users would interact with the multiple modalities of sketch and speech within a virtual environment. In essence, the speech modality provides a semantic component that is comparable with keyboard keywords or textual tags frequently used within traditional 2D interfaces. However, in contrast, a more difficult challenge arises when users are allowed to speak freely. Commonly, this will result in a statement in the form of a sentence, possibly open-ended. This approach could be viewed as allowing in a web search a user to type a sentence as opposed to a few keywords which we have been trained to use for effective searching. However, while immersed in VR, there exists no such pre-training. We, therefore, firstly investigate the question of how many words (grams) are required for effective voice search within the virtual environment when joining works (e.g., the, it, and) are removed. Then secondly, we investigate how an iterative speech and sketch system would work and the effect it would have on the user's search performance.

4.1 Introduction

HCI aims to create practical as well as usable systems. The usability of a system is characterised by the easiness to learn and remember to use, effectiveness and efficiency, enjoyability, and safeness. Three components are evoked by the name HCI: the user, the computer, and how they cooperate. "User" is the human component, which can be a single entity or more entities that collaborate. Human sensory systems such as touch, sight, and hearing are stimulated to improve user experience. This aspect is of fundamental importance as it represents a common background, while other interaction models can be affected by people's knowledge, culture, mental disposition, nationality. Computer components can be desktop or laptop computers, mobile or tablet devices, virtual or augmented reality apparatus, haptics.

Some everyday human actions are speaking, gesturing, and drawing, or sketching.

Voice, gestures, and sketches exhibit relevant and complementary information for the identification of a target object in a large collection. However, to-date it is challenging to integrate and exploit the benefits of the different facets of these inputs.

Nowadays, in some areas, complex interfaces are still dominant because they provide a large number of functionalities. With the advent of deep learning, however, combining simple interfaces with data-driven algorithms has encouraged researchers to investigate interactions such as speech, gestures, eye gazing, etc.

In the previous chapter, we defined a pure sketching interaction to retrieve the target model from a collection of 3D models. Despite the efficiency of the method, this technique does not achieve good results with all the chairs. For example, when the chair is particularly challenging to depict, or a large number of similar chairs are present in the dataset, the user experiences difficulties in exploring the space and finding the right model. Different techniques are applicable to tackle these issues such as improving accuracy with a different inference model, or providing better training to the current model, or even enrich the user interface to support more functionalities (erase, fill tools). On the other hand, differently from an SBIM software running on a 2D display, where the user sketch using a pen on a pad or using the mouse, immersive reality aims to involve all the senses and use information coming from different sources generated by the user voluntarily or involuntarily. For example, including other input mechanisms such as eye gazing, gesturing, or vocal inputs can improve precision, albeit it would require a more complex algorithm to treat the data. In this chapter, we tackle the problem of finding a way to improve the accuracy of the implementation of the system we have defined in the previous chapter. Our purpose is to maintain a high level of user experience, focusing on intuitive interfaces. Testing new neural network models is a simple exercise of replacing the current back-end with a different one. In the last years, many neural network architectures were designed in the continuous search of increasing the accuracy in classification or regression tasks. The naive strategy is to select the deep network with the best accuracy achieved in classification tasks and plug it in place of the current one.

Data augmentation is a powerful technique that extends training data for the model and improves classification [187, 188]. In our case, augmenting data can be achieved in two ways, at evaluation time or training time. The first is increasing the number of camera positions around the 3D object and the sketch. This method is applied during descriptor generation for the models and the sketch. A possible drawback is that each new snapshot needs a CNN for being processed during the interaction stage, and

many images can delay the system response. The second method is generating rotated versions of all the images of the chairs and using them during the training process.

Introducing additional controls in the interface in this virtual environment can improve the user experience but simultaneously can increase the complexity. This mechanism forces the user to practise in order to remember all the functionalities, increasing the learning curve. Moreover, the developer needs to embed these functionalities converting a simple interaction into a multi-status interaction.

An alternative approach is to attach a hand-free input as a supplementary source of information. Using an additional input channel that does not engage the hands, that can operate independently from the position of the body, and that does not need a change of attention by the user, raises much more interest. In this chapter, we tackle the accuracy issue by implementing a multimodal interface inside a virtual environment that extends the preceding version of the VR sketching application. We define the main requirement that is having additional intuitive and natural input. At the same time, this channel has to describe features of a 3D model, and we opted for speech interaction. Speech interaction allows the user to use voice to compose sentences that will be analysed and used for triggering functions. In our study, we focus on the descriptive information of the target chair. Therefore, via a vocal recognition process, the system converts the speech component in a textual description that is used to generate a feature vector. This descriptor can be used individually or in combination with the visual descriptors introduced in the previous chapter to improve the searching task.

Sketch retrieval is affected by the elements in the database. We notice that our sketch retrieval system has degraded accuracy when searching a chair among a group with similar features but with colours applied differently onto the chairs' parts (e. g., two chairs that are identical a part that colours swap between arms and legs). We exploited this weakness and we created a database where the simple sketch system fails to search for specific visual characteristics, and we enriched it with textual meta-information about the style and the shape of the chairs. In this way, we have generated a basis for a descriptive research of the models.

We compare three modalities of interaction in virtual reality for 3D model retrieval from a database: vocal input, sketch input, and the interaction that combines sketch and speech in the same session. The sketch interaction is implemented using an optimised version of the system introduced in the previous chapter. Speech can be very complex to interpret for many reasons that we describe in Section 4.3.1. We opt for an implementation that does not process the full speech but explores the potentialities of this channel. The speech is interpreted by an operator who listens to the vocal query and uses a software interface that selects a subset of models to match the user's request. All the

modalities allow an incremental search that refines the result as the interaction proceeds. To determine how many words need to be considered during the vocal interaction, we ran an initial user test and analysed the optimal number in terms of the rate of success and the time required to complete the search task.

The main contributions of this study are three. The first is the creation of a large variational database with segmented chairs. We notice that 3D sketch-based retrieval does not provide efficient navigation in such a collection. The second is the design of a multimodal interface in a virtual environment where the sketch interaction is integrated with the vocal interface. The resulting interface benefits from the characteristics of the two types of search. We design our multimodal system to provide a translation method between the two types of queries, fostering their integration in a search pipeline. Third, we perform a user study for evaluating the performance of the multimodal interaction and individual interaction modes. Our analysis highlights the strategies adopted by the users that exploit the queries' integration the system provides.

This chapter will continue with Section 4.2 that introduces multimodal interaction. Section 4.3 describes both the novel chair database. Section 4.4 details the unimodal and multimodal interfaces. Section 4.6 shows the user study that compares unimodal interfaces with the multimodal interface and evaluates the outcomes. Therefore, we conclude in Section 4.7.

4.2 Multimodal interaction through sketch and voice inputs

Multimodal human-computer interaction (MMHCI) involves different disciplines such as computer vision, artificial intelligence, NLP, psychology, and many others. In the last years, the development of hardware technologies at an accessible price and the improvement of unimodal interfaces boosted research on MMHCI. The main challenge in a multimodal system is merging different interactions to achieve performance in specific tasks and simultaneously provide a pleasant experience.

It has been established that multimodal interfaces have a lot of advantages: prevention of errors, robustness, errors' correction, improved communication. In addition, they add alternatives and more efficient methods. Oviatt [189] has demonstrated that error-prone technologies can compensate each other instead of increasing redundancy. On the other hand, if the system is not well designed, the use of multimodal technologies could hinder task completion. In this context, Oviatt showed some errors that arise when a user operates through voice interaction. For example, in a multimodal interface consisting of an integrated pen and voice inputs, if the system is designed to work as a speak and point process, it will fail in a large number of functionalities provided to the user. The design of a modal interface must

take these issues into account and at the same time, seek robustness, accuracy, and improved flexibility to be adapted to different tasks.

The correct interpretation of the signals, both visual and auditory, and their timing is crucial. For our study, we consider two input channels integrated into our multimodal interface: sketch and voice. Sketching is one of the most natural methods to represent information in human-to-human communication. Sketching, as a searching method, comes with intrinsic limitations. The super-simplification of the sketched object and its iconic nature make it challenging to identify the exact referenced instance. Nevertheless, in recent years, SBIRs have attracted attention due to the performance that they achieved. Many classic unimodal interfaces use 2D sketches as visual queries to search for objects within a database, and there have been numerous attempts to extend this functionality in virtual environments. Despite promising results, the efficiency of these methods in a context of retrieval depends on the type of database. The features extracted from the elements of the collection affect the capacity of the algorithm to perform a high precision search. The voice, and in particular the language, is a powerful tool that human being has at his disposal to communicate. The use of the speech as a human-machine interface has a very long history. It is used to translate vocal expressions as text and to interpret the text and follow a particular action. The widespread use of vocal user interfaces (VUI) is recent. We can find them in cars, in-home automation systems, operating systems and home appliances such as washing machines and microwave ovens.

However, searching within a database of models through the description of the model is still a challenging task: the model must have associated textual attributes. Firstly, such attributes need to be understandable to all the users, and secondly, a possible search could lead to a large number of potential candidates, degrading the user experience. We believe that a combination of sketch and voice with a properly designed interface increases the explanatory power of the user by acting in a complementary way during the search in a database.

Our study of vocal interaction focuses on the semantic content of the input from the user. Therefore in the user test, we developed a system where the operator acts in-the-loop and operates through dedicated software to send the results of the user's voice query. We describe the study of this part of the system detailing the interface provided to the experimenter.

4.3 Database Design

In Chapter 3 we utilised the ShapeNet database that provides significant variation in terms of visual appearance, which is suitable for testing the sketch modality. However, ShapeNet lacks in several ways

mainly the semantic tags it contains, but also the inter-model variation – the colour and texture of the components of the objects. The 3D Sketch-based retrieval system described in the previous chapter, shows good results in the search task, but when colour inter-model variation happens, a subset of the whole collection may be excluded from the results, or very difficult to navigate. Therefore we are interested in creating a dataset that exhibits significant intra-object variation and semantic tags. We develop the Variational Chair ShapeNet (VCSNET) database in Section 4.3.1, where we outline the automatic generation of the dataset.

4.3.1 Variational Chair ShapeNet (VCSNET) Database

ShapeNet is a collection of chairs with a large number of different shapes, colours, and textures, that do not contain the shapes combined with all the colours present in ShapeNet itself. This choice reduces the cardinality of the chair set drastically and also the possibility, described in the previous Section 3.3.6, that the neural network does not select the target with the colours paired with the right parts of the chair. All these motivations represent weak points of sketch-based search. A collection with a limited number of shapes with all the permutations of colours can manifest these inefficiencies with more evidence. In addition, ShapeNet does not have an extensive collection of textual information. The lack of metadata, the vast heterogeneity of shapes and styles make the analysis of a possible user test intractable.

As our purpose is to improve sketch-based search with speech interaction, we need to cope with the requirement of having some metadata associated with the chairs. A preliminary analysis about text description of a chair shows that the chair can be depicted considering a general style (colour or shape) or a specific component style. This consideration leads to a dramatic boost of information associated with each model, that can proportionally increase with the number of parts of a chair.

When searching a chair verbally among a collection of chairs, language complexity plays a fundamental role. Free speech is hard to analyse because many variables influence a simple description of an object. Some of them are the educational level of the speaker, the language complexity, the contradictions during the query, terms that clash, the mood of the speaker. For example, the sentence "I saw a man on the chair with a telescope" can rise to different interpretations equally valid as seen in Figure 4.1.

To increase the level of control during the vocal queries, the meta-information attached to the chairs needs to be coherent and less ambiguous as possible among all the shapes. On the other hand, we need a limited set of colours avoiding the user to identify a specific colour among the continuous spectrum of colours.

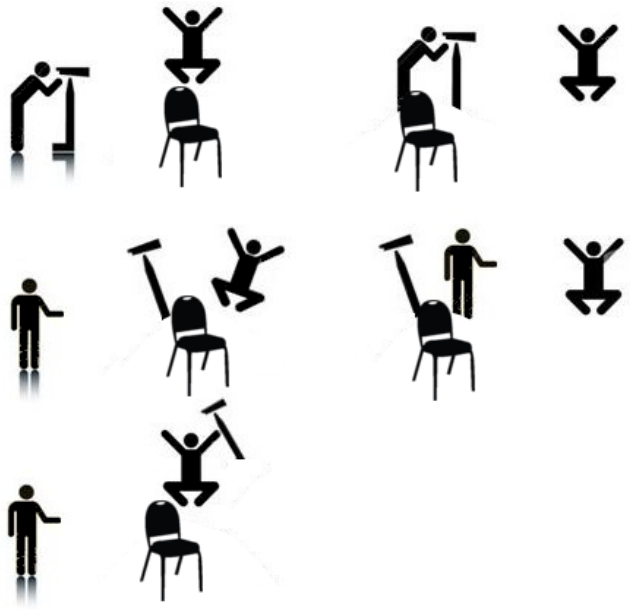


Figure 4.1: "I saw a man on the chair with a telescope" interpretations. Readaptation from the image taken from [7]

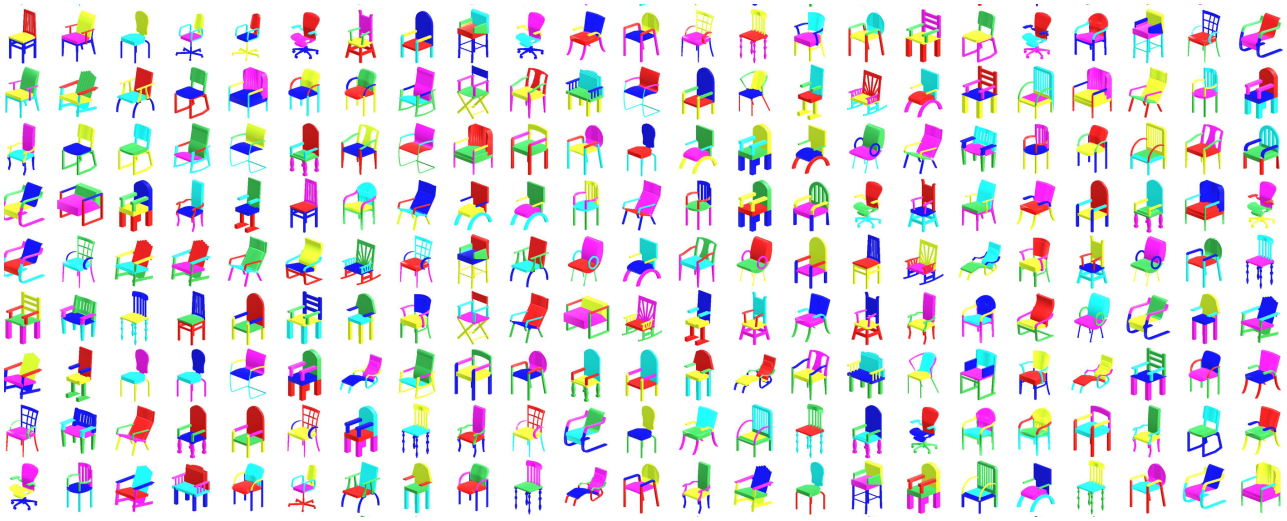


Figure 4.2: A section of VCSNET where 45 different shapes from ShapeNet are collected, then segmented into 4 parts (seat,back,legs and arms). A permutation without repetitions of 4 colours over 6 is then applied to the parts.

Thus, we designed a database that allows us to control the variation of shapes and colours involved. In addition, we provide metadata that permits us to associate concepts to the chairs as a global feature or a feature belonged to one or more of their components. Firstly, we enumerate the chair parts: legs, arms, back, and seat. Therefore, we selected from the ShapeNet collection 45 shapes with a reasonable amount of characteristics that are shared between them and segmented them to have (when present) the four components mentioned above. We reduced the palette to only six colours (red, green, blue, magenta, yellow, and cyan). We motivated this choice with the complexity in describing the exact hue

and saturation of a colour verbally. We permuted them along with the four components of the chairs, avoiding colour repetition, obtaining a total number of colour variations for each shape of 360. Our generated collection contains a total number of 16200 chairs.

To handle textual information used for speech interaction, we generated a dictionary with the words that identify concepts suitable to describe chair or chair component characteristics. For every chair and every chair's component, we created an instance of the dictionary. We associated the concepts in the dictionary with a number from 0 to 100 to describe the relevance of that feature in chair depiction.

4.3.1.1 Chair Shapes and Colours

This section shows the shapes (45) of the chairs we include in the dataset, in Figure 4.3, and for one chair all the colour permutation (360) in Figure 4.4 for a total of 16200. This dataset is 5 time larger than the version used in the previous chapter.

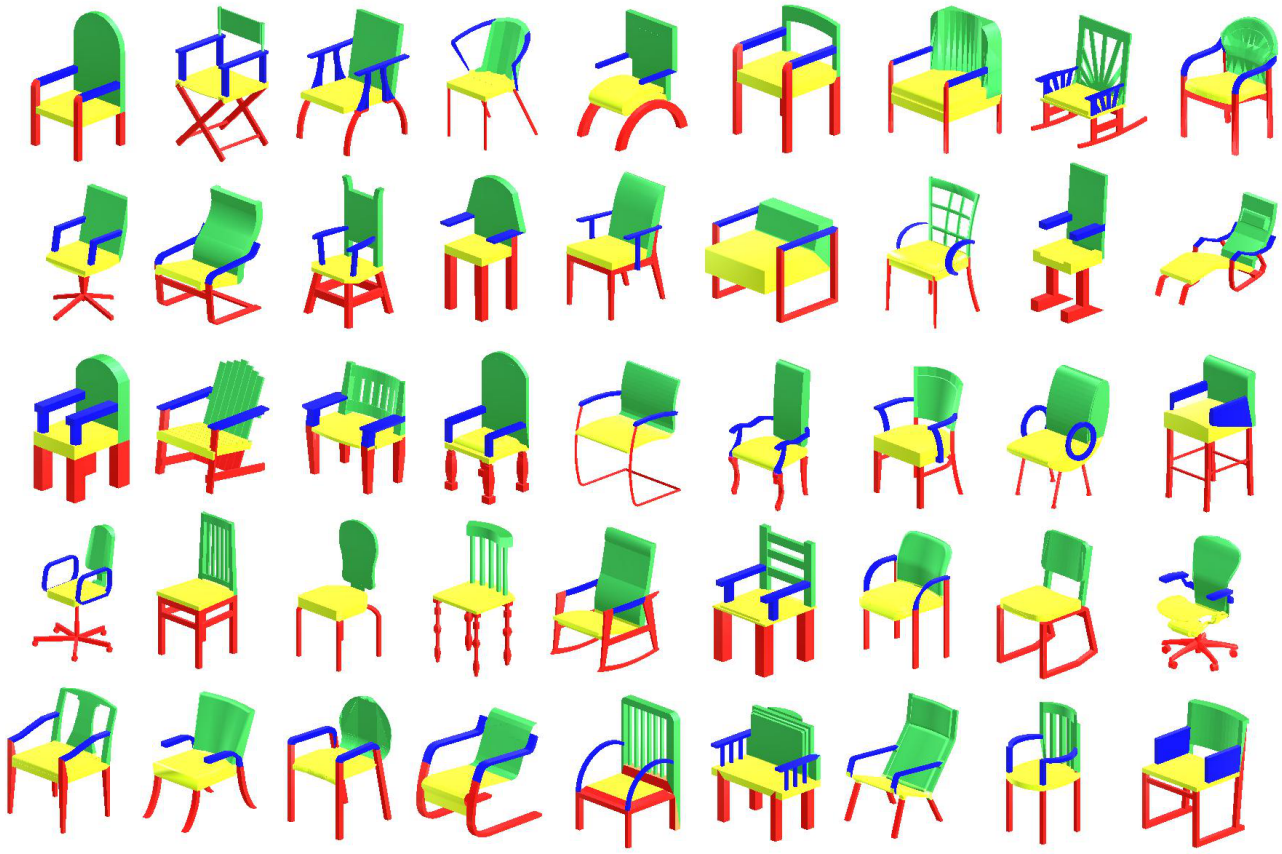


Figure 4.3: The 45 shapes selected for the chairs in the dataset.

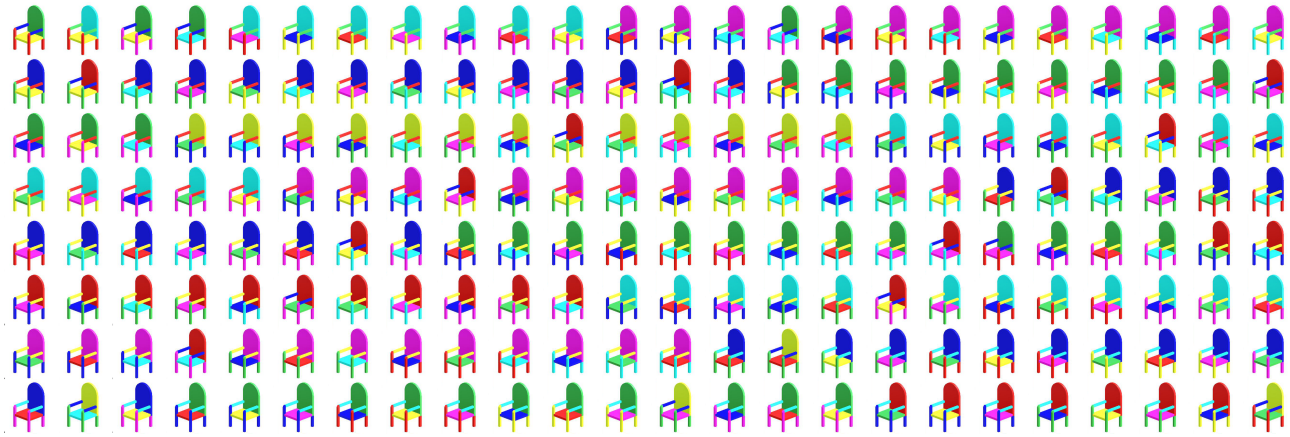


Figure 4.4: The 360 permutations without repetitions of colour in one type of chair.

4.4 Modality Interface Design

In the following section, we describe the interaction methods for our retrieval system, including the back-end. For all the modalities, the user is immersed in a virtual environment and can interact with the system with the voice and/or with mid-air 3D sketch (as shown in Figure 4.5).

Voice User Interface Predicting the words and the sequence of queries for a 3D model searching task is challenging. A possible workflow for pure voice interaction is describing the chair with a long sentence and without any pause. This description would result in a single query with a lot of information and hard to translate into the suggestion of a single model. Our approach is to split this sentence into more queries each one with a limited amount of information that can be treated more simply. In this way, eventual word clashing, or contradicting queries can be treated separately. If some issues occur in the sentence (such as concept misunderstanding, lossy translation, incomprehensible words), only a small portion of the information is rejected. We design an interface and put some constraints on the user, in order to cope with some notorious language complexities: syntactic ambiguities, semantic ambiguities, implicit subjects, positive or negative queries.

We design a user test where the participant interacts with the system, believing it to be completely autonomous, while it is partially managed by the experimenter (Experimenter-in-the-loop). This experiment is the so-called Wizard of Oz experiment (as shown in Figure 4.6), very popular in experimental psychology.

The experimenter is instructed to consider only the relevant words and classify them in the category defined in the dictionary used for the feature vector. These actions are equivalent to tokenisation, lemmatisation, and stemming for a NLP library. In the desktop application, they apply all the changes coming from user speech, clicking buttons associated with those features incrementing or decrementing

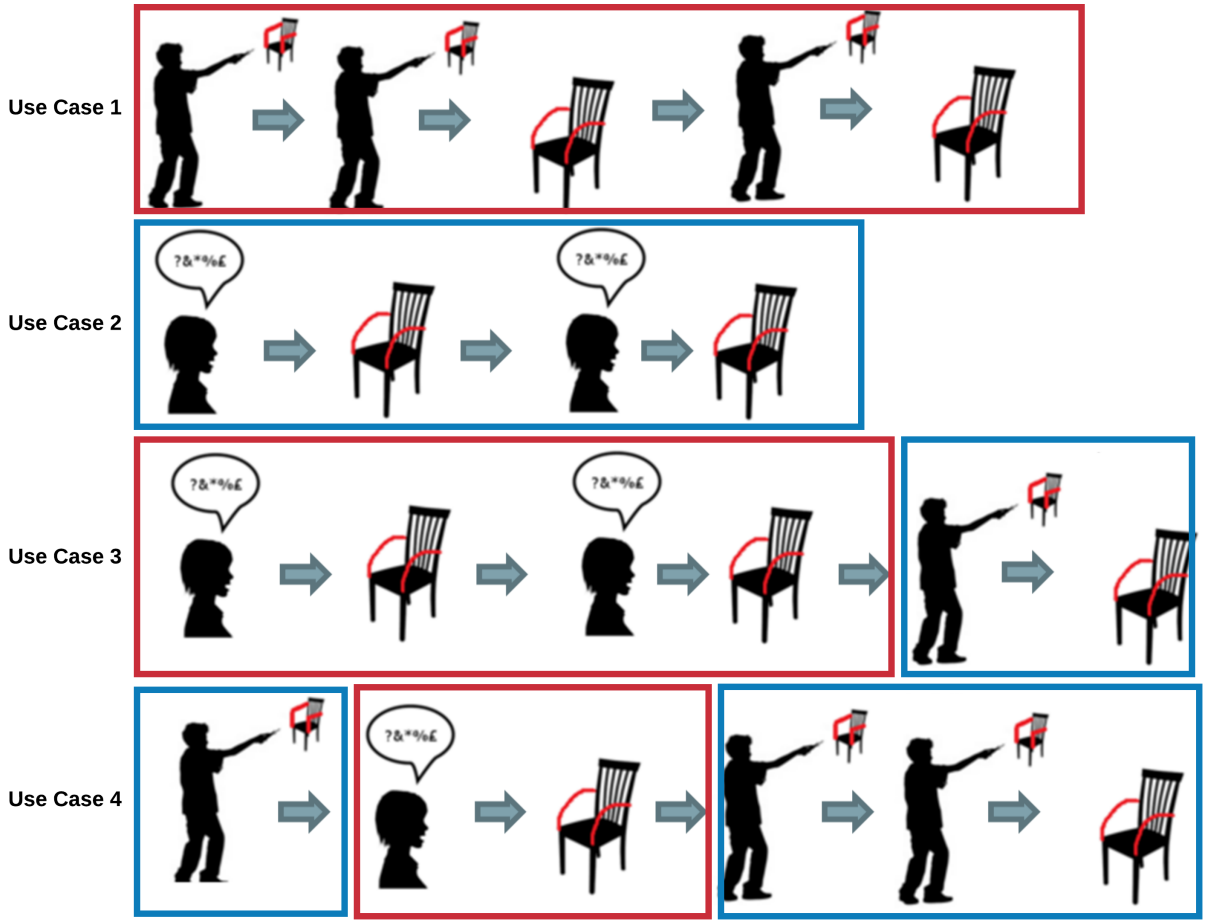


Figure 4.5: This Figure shows a selection of different use cases. Red boxes represent sketch queries, blue boxes vocal queries. The first two use cases exploit mutually sketch or vocal interactions, the third and the fourth use cases a combination of them.

values in the feature vector. In this way, we can count the words in a sentence and consider the only relevant word for the count. We propose three sessions where the queries differ by the number of words: one with bi-grams, one with four-grams, and one for six-grams. Our purpose is to discover which n-gram achieves the best accuracy in terms of the number of found models. Therefore, we run the experiment recording the final model selected by the user, the time needed to complete the search, and the used words. We avoid a direct selection of the chair by the experimenter that he can be influenced by knowing the target. Therefore we designed an external software that includes a weighted random process of selecting the result chairs to propose to the user.

We opted for this solution because the complexity of the task can not be managed by software easily. Deep algorithms such as Recurrent Neural Networks (RNN) are able to deal with functions such as sentiment analysis or text analysis. However, having a completely autonomous system added multiple sources of errors, as well as a considerable amount of additional work. Firstly, voice

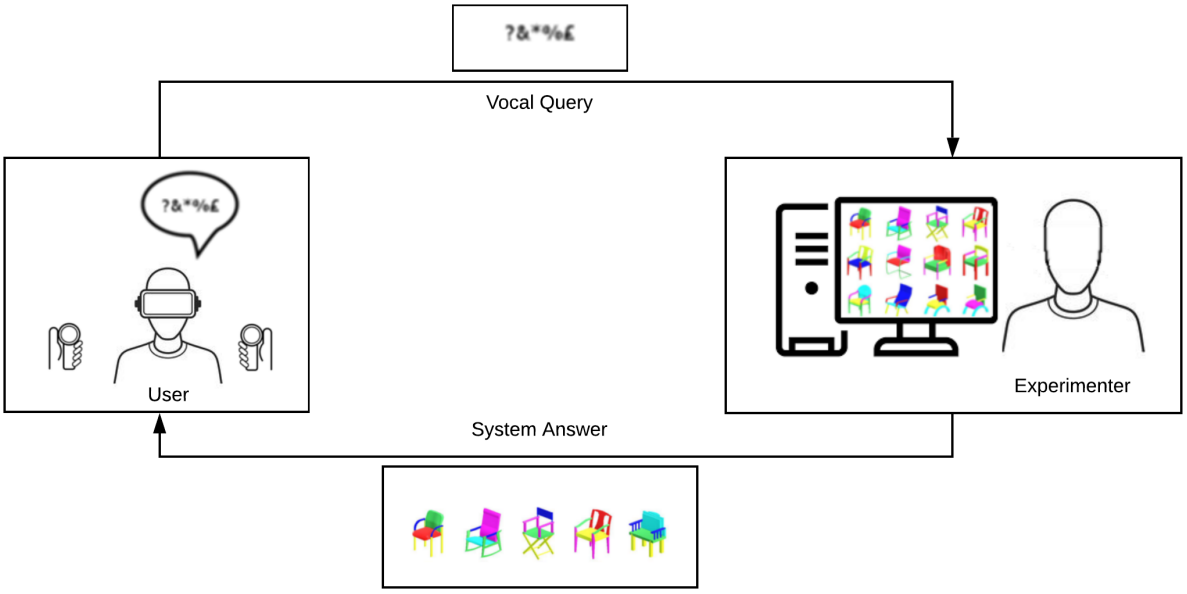


Figure 4.6: Wizard Of Oz experiment: the user on the left interacts describing the chair, while on the right, the experimenter uses an interface that selects the best five chairs from the dataset according to the user's description.

recognition would have been essential to convert the participant's voice to text, and secondly, an eventual RNN needed to be trained with chair descriptions. These two components can add errors or delay in the pipeline while an experimenter could handle it in an effortless way.

The user is immersed in a 3D environment consisting of a furnished room and a floating GUI where the search results are shown. The user verbally describes the chair proposed as the target, visualised as an image on a panel attached to the GUI. The user can detail the chair using the dictionary of concepts permanently showed in a transparent layer in front of him. All the synonyms and contraries are allowed, and each query terminates with the word "stop". The experimenter is using desktop software that provides a simple interface for selecting the features described by the user, increasing the score associated with a specific concept in the dictionary. The task of the experimenter is to build the feature vector and send back five different models as the result set. The software performs the search via Euclidean distance with the feature vector generated for all the chairs in the collection. The user will visualise these models and will perform a selection accordingly with the chair closer to the target. When the chair is selected, the 3D model appears in the 3D environment. This process is iterative until the user is satisfied with the selection.

Sketch User Interface We based our sketch interface on the description in the previous chapter. We provide small modifications to the system for generating the snapshots for the multi-view process,

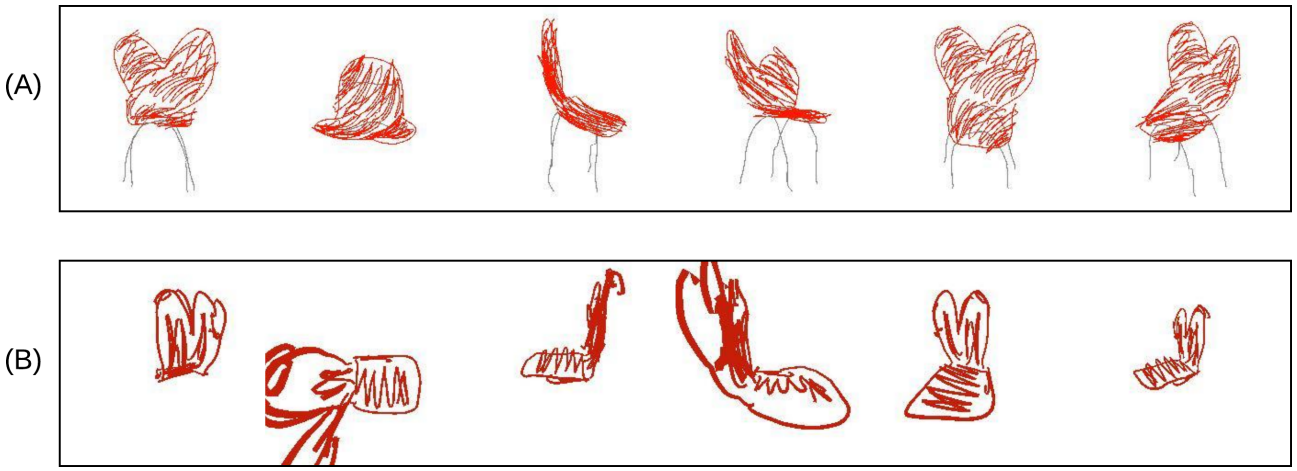


Figure 4.7: The difference in the snapshots taken from diverse camera types: (A) shows the orthographic camera snapshots while (B) are the snapshots from a perspective camera.

changing the camera type. We pass from perspective camera to orthogonal one, improving the camera positions and field of view before generating the snapshots. These changes improved the input for CNN by framing sketches and models correctly, without cropped elements (see Figure 4.7).

The software inherits the same house's room described in the previous chapter, as VR environment and used the same CNN back-end to interpret the sketches. The user can trigger the system both with the sketch or with the sketch and the model. The results' panel contains only five chairs with different shapes.

Hybrid Sketch-Voice User Interface When designing the hybrid method that mixes up the two interaction types, we needed to formalise the definition of queries in order to avoid combination issues during the test. The session in which we allow the participants to use both voice and sketch required one important constraint: the participant can use one method at a time without overlapping them but with the possibility of creating any sequence of queries to perform the search. Thus, sketch and voice query should be organised with the same workflow to avoid incoherent states. For example, if the current version of the chair is not defined, the incremental search becomes impossible. With this precondition, we propose a definition of query which includes an input part, a processing part, and a selection part. For the sketch, the input is the snapshots, and the process part is determined by the CNN back-end.

The selection part is the final stage where the user is selecting one of the chairs present in the resulting set (as shown in sketch query red box in Figure 4.8). The speech terminating with the word "stop" is the termination of the voice query. The processing part is the experimenter-in-the-loop that generates the features vector. The selecting part is the same as the sketch query (as shown in speech query blue box in Figure 4.8). The query is valid only if input, processing and selection are present,

otherwise, the query is rejected. This formalisation allows the system to concatenate different queries. We avoid errors like an undefined current selected chair, which is the major problem during an iterative search using different channels. The cyclic graph in Figure 4.9 describes the possible workflows during the hybrid test. The software implementation, with these preconditions, is implemented in VR application for the user and desktop application for the experimenter.

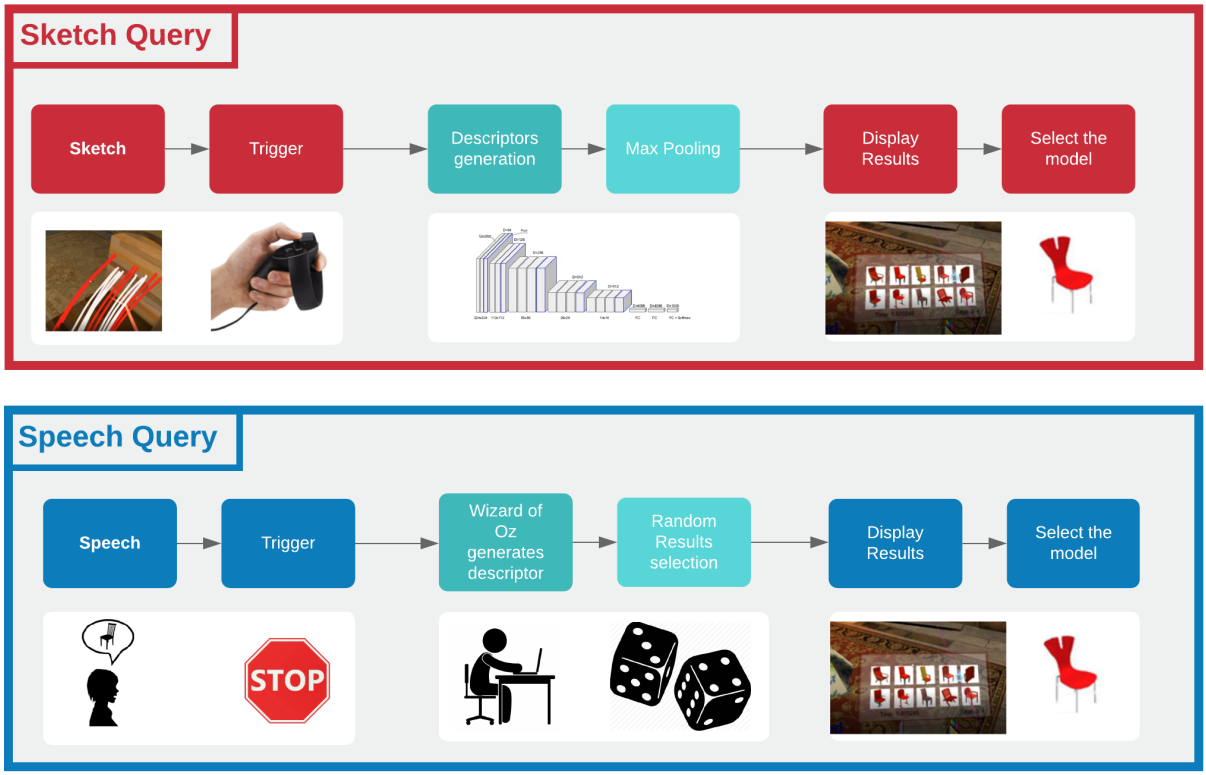


Figure 4.8: This diagram defines the different stages that formed sketch and speech queries. Both the query types include an interaction phase, a processing phase, and a display phase.

4.5 Wizard of Oz experiment

In this section, we motivate why Wizard of Oz (WoZ) procedure is the most suitable for our study when speech interaction is combined with the sketch (Section 4.5.1). In addition, we exhibit both the dictionary and chair collection we automatically create by using colour permutations (Section 4.5.2).

4.5.1 Why do we choose a Wizard of Oz approach?

Here, we briefly describe the state of the art of NLP algorithms. After enumerating the different steps of our speech interaction pipeline, we highlight how they can be implemented automatically and show the pros and cons.

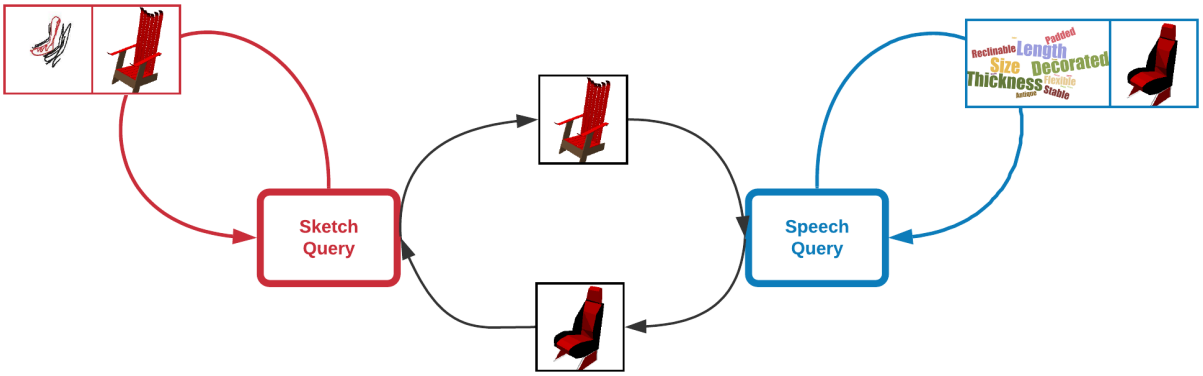


Figure 4.9: The graph describes the sequences of queries during a searching session. Red arrows refer to a sketch search where the user using a 3D draw, and a model performs the query. Blue arrows refer to vocal queries where the user describing the chair and selecting a model performs the query. The central part of the diagram shows the connection between the queries where a selected model is the input for the next query.

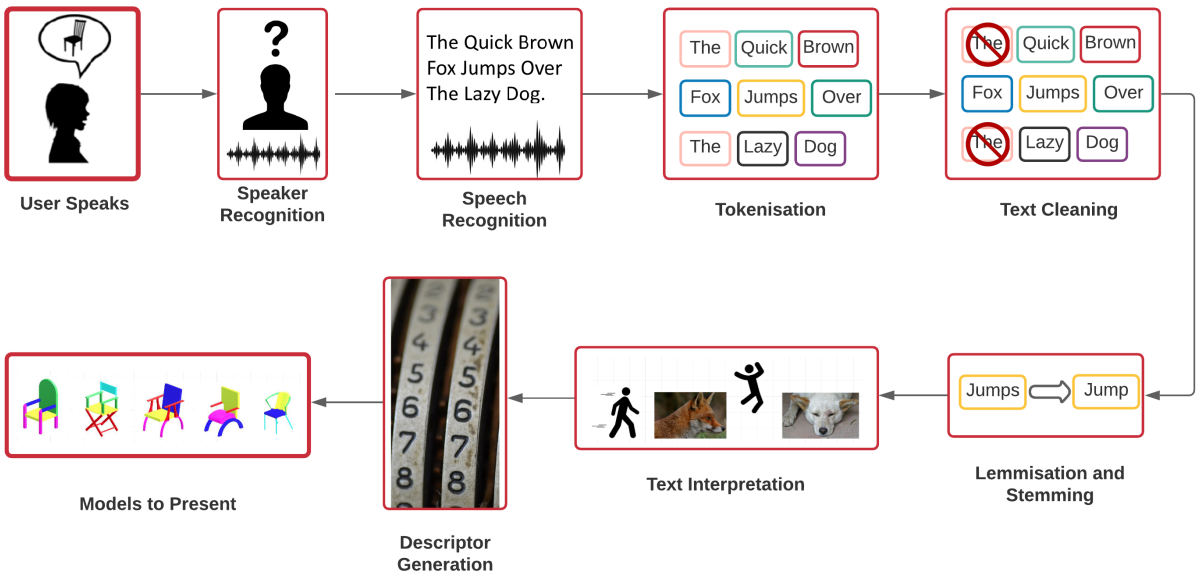


Figure 4.10: This diagram shows our speech pipeline. The required steps are speaker identification, speech recognition, tokenisation-lemmatisation-stemming, text interpretation, descriptor creation, and results selection.

4.5.1.1 Natural Language Process State of the Art

Recently, deep learning techniques produced promising results in the context of text processing, such as sentiment analysis or semantic extraction. One of the first attempts was Recurrent Neural Networks (RNN) that are applied in text analysis [190]. Pascanu et al. [191] proved that such family of algorithms suffer from vanishing gradients. This means that they can ignore part of the sentence (also important for the meaning) and forget terms. More recent works such as [192] uses Long Short-Term Memory (LSTM) and its variants to tackle the memory issue, improving the model in remembering data. Moreover, Google Brain and Google Research recently have developed mass parallel processing implementations that foster attention mechanisms, resulting in Transformer architectures [193]. The improvement achieved in 2018 by deep learning in NLP can be considered equivalent to the influence that ImageNet [194] had in 2012 for computer vision. At first, we consider defining a full-automatic pipeline with a defined software stack. We detail this pipeline in the following section, showing the pro and cons of each element. After analysing the advantages and disadvantages, we concluded that an automatic pipeline injects errors at different stages. These errors accumulate and tamper the final result. For this reason, we opted for a semiautomatic approach with the experimenter-in-the-loop. Detail motivation follows in the next section.

4.5.1.2 Requirements for the Speech Interaction Experiment

Our experiment needs audio input that is participant speech. This information can be recorded directly with the Oculus Rift device, which integrates a microphone in the headset. Therefore, each session produces an audio file Mono, PCM 11025 Hz.

In the second experiment (4.6.1), we compare three modalities. While sketch interaction is the one we described in the previous chapter, speech interaction needs to be defined and integrated into the system. When we analysed the requirements, we split the speech interaction pipeline into stages. The pipeline aims to transform vocal input into a speech query and then into a chair subset that is the system's proposed results (as shown in Figure 4.10).

The stages (or steps) of our pipeline begin with the user that describes by voice the chair model and terminates with a small collection of chairs presented to the user in VR. The steps between these two perform a specific task, and the total time of execution must last a maximum of 5 seconds. Following we enumerate and describe the steps:

Voice recognition Recognising the speaker is the process that identifies the active speaker from the audio generated. During the experiment, the user is alone with the experimenter and can interact, for example, by asking for elucidations. In this context, it is important to know who is talking as Oculus

microphone can also capture the voice of the experimenter that can interfere with the correct input. A possible alternative is to avoid such interaction and let the user interact only with the machine.

Speech recognition Speech recognition (or voice dictation) aims to develop methodologies that are able to recognise and translate the speech into text. Despite having high accuracy is a challenging task, dictation software usually can be trained with the speaker's voice. We selected three different services that do not require a training phase, and we tested them with the file recorded with Oculus microphone. This stage is crucial because a low accuracy can inject the wrong information into the pipeline. Another possible issue of this stage is to introduce words expressed by the user but not relevant for the search, such as questions or comments.

Tokenisation Tokenisation aims to converts the text into single elements (tokens). Each token is a sequence of characters that represent a word of the used language. This process is quite simple and implemented in NLP libraries.

Text cleaning Text cleaning aims to filter the list of tokens, removing a subset of them. The removed objects do not have relevant information, and NLP libraries have dictionary-based implementations.

Lemmatisation and Stemming Lemmatisation and Stemming are two different processes but with the common purpose of reducing the number of tokens, grouping them and using only one representative (with the same root) for that category. Lemmatisation works on the morphological analysis of the word while stemming remove prefixes or suffixes of the word.

Text interpretation Text interpretation aims to extract meanings from the sequence of words. Groups of sequential words, in our case, describe the chair's attributes and this information needs to be extracted and stored in a descriptor (feature vector).

Models' selection In this final step, the system uses the generated descriptor to produce the most similar chairs in the collection, by using Euclidean distance.

4.5.1.3 Pipeline considerations

In this section, we discuss the issues that can be introduced by each pipeline's step. Each pipeline step can be implemented as an automatic process. Both automatic and semiautomatic implementation (with a human in the loop) need to be analysed to decide the best solution.

We required speaker recognition as, during the test, the participant can ask the experimenter some information about the search or some clarifications. The question can be very different, and the experimenter's answer is recorded by Oculus equipment even if with a lower intensity. His answer contains input that can be considered noise, and so it needs to be removed. To prevent such noise, we

can instruct the user to avoid talking to the experimenter, and this stage could be considered optional. If automatic, this step needs to have 100% of accuracy because additional words from a different source can compromise the query generation. If handled by a human, this stage is managed without difficulties considering only the input from the user.

A voice dictation software performs the conversion from speech to text. The most important metric, in this case, is the accuracy of the software that performs the transcript. Moreover, we noticed that the participants sometimes speak with them-selves providing not a relevant information, or react to some events that happen in the virtual environment: commenting the results, self-questions about how to sketch and so on. This additional information needs to be rejected by an automatic system because it is a noise source. We tested three speech-to-text implementations without fine-tuning the software with the speaker's voice, replicating the condition of the experiment. In Section 4.5.1.4 we show the accuracy of each service, that is unsatisfactory for all of them.

Excluding part of the speech not relevant, a possible solution is to implement a flag in the VR software that enables or disables the speech recognition. This option could be triggered with the hand controller, violating in this way, our intention to have independent input channels.

On the other hand, an experimenter can not convert speech to written text in real-time but can be helped by an intuitive interface and instructing the participants to follow a dictionary of meanings. In the VR software, the panel with all the meanings is always present and visible. This dictionary includes all the concepts valid for the chair collection. Moreover, we determine a speech query length to be 10 seconds maximum as continuous speech is a hard task for both automatic or semiautomatic implementation.

Tokenisation and text cleaning is a process that is implemented in all NLP libraries, and it is easy to achieve with an automatic or semiautomatic stage. In the same way, lemmatisation and stemming can be performed by NLP libraries or by a human-provided with a correct interface.

State-of-the-art models such as Transformer are able to generate the descriptor that contains the features of the input text. As with all supervised problems, training the model is an essential step, and requires chair descriptions coupled with the correct labels. To create such a training dataset, a large amount of effort is required. Although the colour-based meta-information can be automatically generated, the rest of the description needs the human intervention, for example, by using Amazon Mechanical Turk service or by recruiting additional participants. Differently, by providing the experimenter with an efficient user interface, the experimenter is able to generate a descriptor with some clicks of the mouse over the right dictionary concepts when mentioned by the user.

The model selection step, that is the last element of the pipeline just before the presentation, can be managed automatically. All the stages described before may introduce errors. Error estimation is also difficult, and they accumulate step after step. In the following section, we analyse the most important stage of the pipeline: speech recognition stage.

4.5.1.4 Speech Recognition: Speech to Text services

Here, we show the results of three speech recognition services that process the audio coming from some participants that tested our equipment. Ten audio files were processed by the following services:

1. Watson IBM (https://speech-to-text-demo.ng.bluemix.net/)
2. SONIX (https://sonix.ai/)
3. google speech to text (https://cloud.google.com/speech-to-text)

We calculated the accuracy by grouping consecutive keywords in each query. Every group transmit information related to the chair style or colour, or of one of the chair's part. If the participant says "blue curved arm" only that word sequence (or eventually "straight blue arm") is considered a positive score. Watson IBM scored 37% of accuracy, SONIX service 60%, and Google Speech only 16%. These results are disappointing and can be caused by multiple factors:

1. Oculus microphone can inject noise in the system.
2. Each participant has an audio profile made by tempo, rhythm, pitch, and with fluency and accents can impact a lot on the accuracy.
3. We did not train the software with the participant's voice using the default settings.

With the help of the interface and a limited dictionary, the experimenter can manage speech to text conversion easily. In this case, we obtain a reliable semi-automatic system. We discuss the possibility of a fully-automatic interface in Section 6.2.2

4.5.2 Dictionary and Dataset

Our dictionary contains the following concepts that describe a chair or a part of it. We pair each feature with a value that exhibits how much that feature is present in the chair description. The concepts listed in the dictionary are the following: Height-Length, Size, Thickness, Decoration, Curviness, Modernity, Antiquity, Slattedness, Swiveling, Flexibility, Stability, Reclinability, Padding, Slantedness, Canvas, Missingness (as showed in Figure 4.11). All the concepts can be associated with the chair, or to the part of the segmented chairs: back, seat, arms, legs.

Figure 4.11: Concepts present in the dictionary in addiction to the dimensions of the chair.

4.6 Sketch and Voice interaction experiments descriptions

To evaluate our multimodal solution, we designed two user studies. The goal of the first experiment was to find the optimal number of words per vocal query for achieving the best result in the search task of a 3D model inside the database. The second and experiment was designed to compare three different methods of interaction: pure speech session, straight sketch session, and combined sketch and speech session. In this section, firstly we describe the main experiment 4.6.1, and dedicate a shorter section for the second 4.6.2.

4.6.1 User Study: Sketch and Voice interaction for retrieving task

This experiment aimed to compare different modalities to search for a model in an extensive database in an immersive environment. Each user performed three test sessions, one per interaction type: sketch, speech, and the combination of them.

After collecting the suggestions from the users that performed the first pilot (described in Section 4.6.2.1), we applied the modifications to the voice interface in order to improve its quality and proficiency. Firstly we reduced the dictionary from a total of 30 concepts to 20. We purged the terms not used during the tests or the ambiguous words. In particular, we merged words that represent the opposite meanings of the same concept and introduce a few terms that were not present before but were indicated by the users. In addition, we introduced a maximum and minimum value for the feature with a discrete step variation to avoid in-between values or overflows. We framed the concepts of height/length, size, thickness to avoid ambiguities. We created some shortcuts and improved the Wizard of Oz software to speed up the feature vector generation. We created a mechanism that synchronised the current model placed in the virtual room with the existing feature vector for the

speech interaction. In this way, the unpleasant issue of losing some features between two consecutive queries that occasionally happened in the first experiment disappeared in the second. In addition, we improved the visualisation of the words layer in the virtual environment, positioning them closer to the centre of the frustum.

For all the sessions, we used the same virtual environment but accordingly to the proposed method, we enabled speech interaction, sketch interaction, or both. In particular, the speech interaction method is the same as the first experiment with the only exception that we consider the number of words achieved as the best result from that experiment. The sketch method allows the participant to search the target chair using 3D sketching with the same modalities described in the previous chapter. The virtual reality software was developed from previous versions (Section 3.3.3). As in the previous experiment, the scene consisted of a furnished room. The sketching interaction was inherited from the VR application described in the previous chapter. The only difference consists in the palette (6 buttons) that shows only the colour allowed in the database. The user can still sketch on top of the current model triggering the system with a query that carries only the 3D sketch alone or in combination with the current model.

The user has one minute and thirty seconds to find the right chair. We involved 10 participants (different from the previous user test), and each of them randomly performs the three types of modalities in a sequence of 9 searches, over 27 models of the total of 45 in the database. As with the previous experiment, we recorded the success rate, the time to complete the task, a self-graded evaluation of the English level, and a subjective evaluation of the user experience through a final questionnaire.

4.6.1.1 Results for pure voice queries, pure sketch queries and combined voice and sketch queries

6 of the participants were male (4 female), and the range of the age of the participants was from 18 to 43 years. Five of the participants had no experience in virtual reality or tested once in their life.

The rate of success for the three different methods is 29 over 90 for the voice, 0 over 90 for the sketch and 38 over 90 for sketch and voice, see Figure 4.12.

The average time to complete is 80.5 for the voice session, 90.0 for the sketch session, and 81.4 for the sketch and voice. Finally, the average number of tries for the voice interaction is 3.5, for sketch 2.7 and sketch and voice 3.4.

In addition, we measured the rate of success and the time to complete considering only the shape of the chair as the final target. The rate of being successful, in this case, shows an improvement for voice of 32 over 90 and for pure sketch session the right target shape is selected 8 times over 90 (see

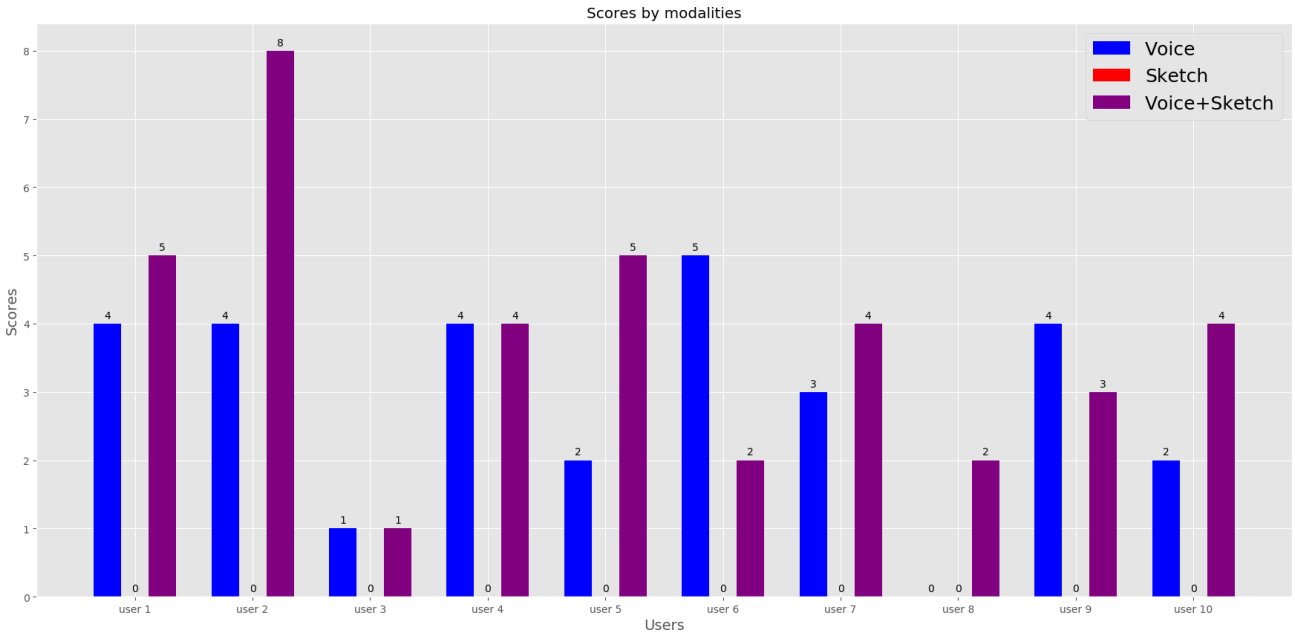


Figure 4.12: This chart depicts the number of successes (values on top of the bars) in finding the target. On the X-axis, the users are displayed with their results in each method. On the Y-axis, the number of successes for each method. It is immediately visible that with the sketch method, the users are not able to find the correct target. This result is caused by the difficulty of the system of assigning the exact colour to the exact part of the model.

Figure 4.13).

The questionnaire at the end of the test asked to rate the user experience for the different modalities using a score between 0 (very bad) and 5 (very good). Speech interaction gets an average of 3.7 out of 5, sketch gets 2.4 out of 5, and sketch and voice 3.6 out of 5. We asked participants to rate the performance of the system with a score between 0 (very bad) and 5 (very good), achieving an average of 3.1. In addition, the user was asked to score to evaluate if the retrieval system was completely automatic or a human component was involved. The possible values range between 1 (fully managed by a human being) to 10 (fully automatic), and the final result is 5.7.

We assume that the sketch and voice session will outperform both the voice session and the sketch session. On the one hand, voice query is very exact with colours, describes with a good detail the shape and represent a robust method of searching. However, some shapes are difficult to describe in detail and to extract among a group of chairs with similar characteristic. Therefore, the voice method shows some deficiency that can lead the user also to a frustrating experience. We can identify two situations where the user can not progress in the search. The first case is the impossibility to get a different result from the last query. The second is a continued bouncing between chairs that share some characteristics with the target shape without selecting the right one.

In many situations, selecting the right shape using the voice means a right path to success. The

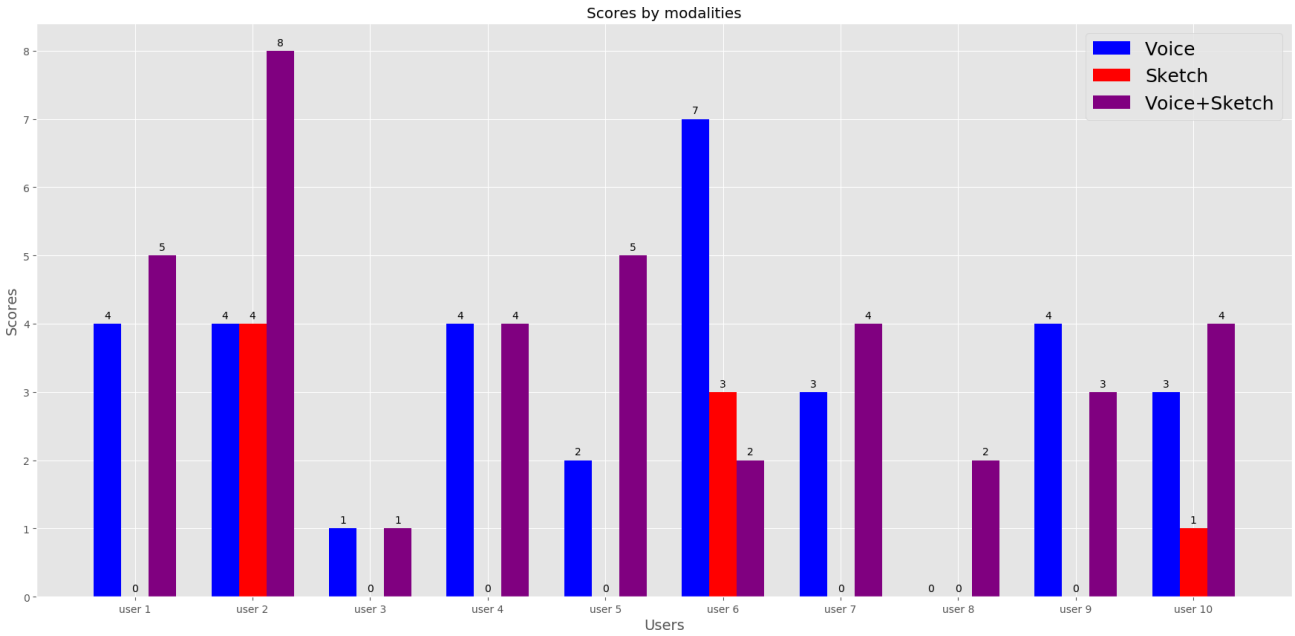


Figure 4.13: This chart shows the number of successes (values on top of the bars) per user and modalities considering only the retrieved shape by the participant. It is visible that sketch contribution appears for some users but still not for others.

only difference between the target chair and the current one is simply the colours that voice query address in one or two queries. Sketch query is less efficient rather than voice query when searching for shape and colour. In particular, even if the shape is found, the results coming from the CNN back-end does not have colours in the right positions (as supposed when designing the database).

The additional analysis that considers only the target shape shows the difference within the sketch session. While the user can often select the correct shape, they cannot select the correct colour. The efficiency of the voice over the sketch is due to the possibility of describing precise characteristics without any error of interpretation. In that case, the sketch is subject to an interpretation that, for the current system, can not ensure a correct correspondence. For features where the text description can be ambiguous or can lead to more than one result, the sketch has the advantage that it can depict them quickly, giving a better portrayal of them.

We noticed different strategies developed by users that emerged from the modalities, as shown in Figure 4.14. For example, in the context of pure voice interaction, the user started searching for the shape, without paying attention to the colour in the first instance. When the query is completed selecting a new chair, this last selection will define the current feature vector. In this way, the next five chairs will have a similar descriptor. This result will appear when the next query will be processed, possibly with an additional description that can be focused on shape and colours. A frequent selection with different selected chairs let the user explore the shape space faster. When the shape is found, the

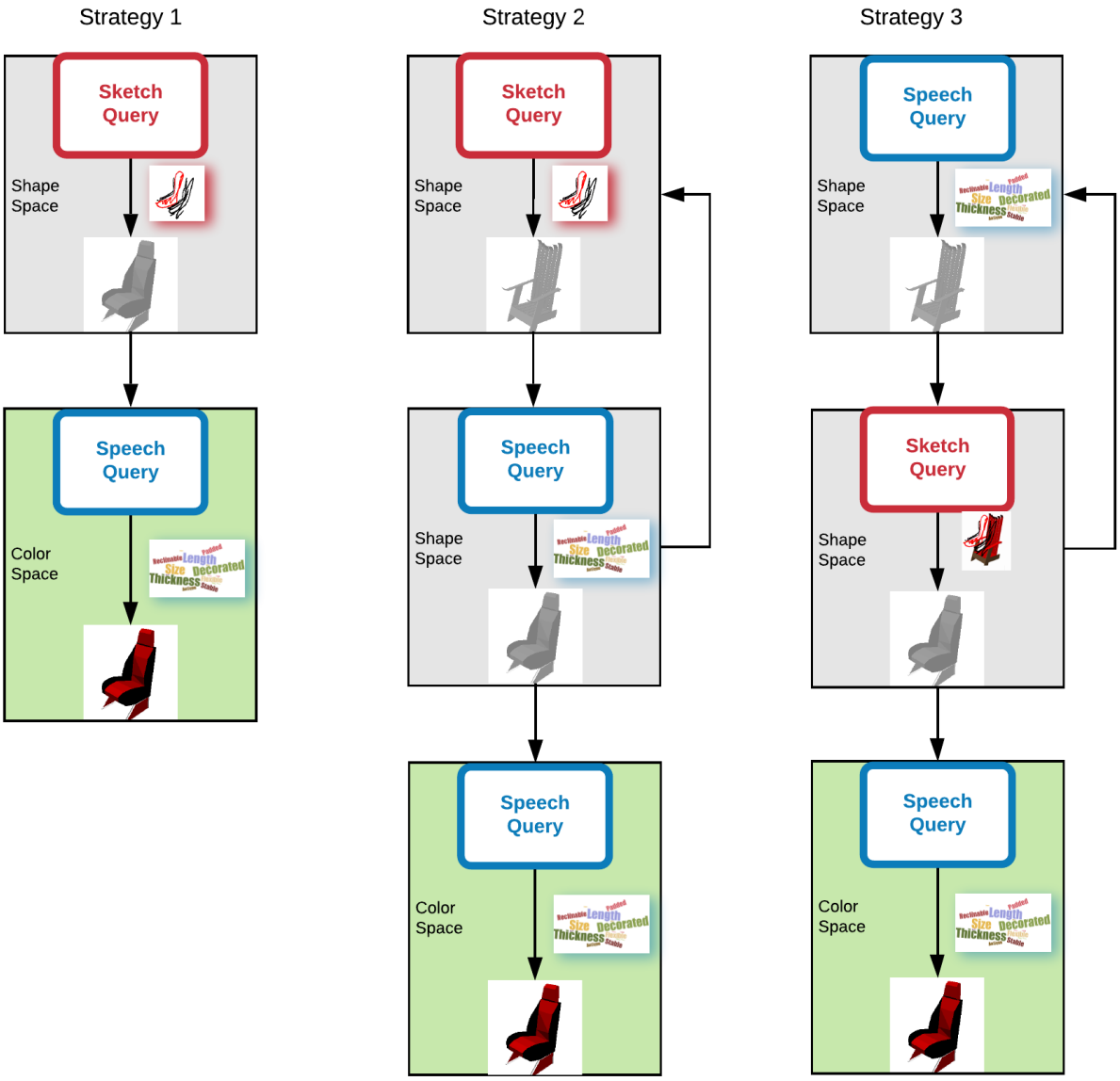


Figure 4.14: After an analysis of the log recorded for each search, we discover some approaches during hybrid interaction sessions that involved both the interactions. The left-most strategy shows a simple sketch query for searching the shape followed by a speech query for the colours, the middle strategy iterate between sketch and vocal query for searching the shape and end with the vocal for the colours. The right-most strategy starts with a speech query to find the shape and iterate with the sketch and ends with the vocal for searching the colours. Sketch strategies to find the colours are unsuccessful.

colours can be described very quickly.

The hybrid solution is the most interesting because of the interaction combination. The user can adopt only the voice interaction, but the most fruitful results are achieved with a sequence where both voice and sketch interaction are used. For example, the user selects the right colours and provides a complete sketch immediately or repeatedly triggers the system with a selected model and additional sketch. In this second case, the user tries to complete some missing parts of the chair or replace the component if the colour does not match. We notice that even if the shape is found in many cases, the colours, in general, is not correct. The CNN back-end is prone to use the same colours depicted by the user but frequently are associated with the wrong component of the chair. The synergy between the two modalities can be very effective as it can merge the most successful patterns of interactions. On the other way, we noticed, that sketch interaction alone is never used during a hybrid session. We can identify three successful primary strategies of queries, with small variations or additional loops.

The first strategy is linear and straightforward: sketching to find a shape, and speaking to find the colour. The second strategy is sketching to find the right chair and select the most similar to the target. The user begins a new search via voice interaction where the starting feature descriptor changed accordingly with the selection from the previous search. This query terminates with a chair selection that is the input for the next sketch query. In this way, voice and sketch can be used sequentially exploring the chair space using two different features spaces. The final step is using voice interaction to select the colours. The third successful strategy starts with voice mode for selecting the shape and then sketch to depict additional elements with the chair selected from the previous query. Also, in this case, a loop between sketch and voice query can start. With this incremental search in both the feature spaces, the target can be found quickly. The final stage is still using vocal description to search for the colours.

The main challenge is finding the shape because the description can be subjective and ambiguous, and the sketch can find difficulties for some types of chairs. The combination of the two modalities tends to improve the performance when taken individually. The difficulties of the single modes are not overlapping, and when merged, the user can select the best query to overcome the issue.

This section that describes the results of our experiment validates both hypothesis [H2.2] that states that “a multi-modal system including sketch and speech, to work properly, needs a formal query definition and a specific interaction pipeline” and hypothesis [H2.3] that asserts that “the combination of voice and sketch interaction improves the search in an immersive context when compared to individual techniques.”

4.6.2 User Study: Voice interaction for retrieving task

In this experiment, we asked the user to retrieve a chair from the database using merely the voice in a virtual environment. The outcome is used in the main experiment. The participant is located in the furnished room with a default white chair that can be replaced by retrieved models. We showed the target chair to the user, positioning an image of it on the GUI attached to the wrist. The user can describe the chair and terminate the description with a stop command (the word "stop") that starts the query. The language used in all the experiment is English. The participant has 90 seconds to complete the search. Given the nature of the voice query input, we ask the participant to express the query using a sentence that describes one or more features of the chair. In addition, we advise the participant that the system can not deal with a long sequence, and the duration of speaking should be no longer than 10 seconds. To reduce complexity, we limited the user dictionary to 30 words, excluding the words that describe the colours and chair components. We instruct the participant to use those words as concepts disclosing any synonyms or antonyms, and comparatives. We recruited 10 participants. We randomised the models during their session, and each user did 27 search sessions. The user performed nine searching tasks with the specific n-gram, and the three modalities were randomised. The user was unaware that the experimenter was evaluating the number of words during the speech interaction. Indeed he/she was instructed to believe that software is interpreting his/her vocal communication. Before starting the experiment, the user was asked to fill a form. In this form, we recorded the familiarity with virtual environments with 3D software or games, from a level of 0 (no-familiarity) to 5(high-familiarity). Using the same level (0-5) We proposed to rate their experience in the virtual reality system if applicable (average 2.5). Therefore, we asked self-rating of the perspective-taking (average 2.8), the orientation (average 2.6), visualisation of 3D objects and space (average 3), and way-finding (average 2.6). We recorded the rate of success and the time required to terminate each searching task. After completing the experiment, we asked the user to fill a final form where we proposed to rate the proficiency of the system if some terms were missing. The best n-gram is used as the constant number of words evaluated by the experimenter in the second experiment.

4.6.2.1 Results: How many n-grams to consider?

We investigated the differences between the three n-gram sessions (2,4 and 6). Before running the experiment, pilots showed that considering queries larger than 6 words were hard for the experimenter. We evaluated our test over the 45 distinct chairs, 27 proposed to each participant, 9 for each n-gram. Thus, the user experimented the 3 n-grams and was unaware of the study type. We recorded the accuracy of the returned model, time, and the number of tries to complete the task. The accuracy was

evaluated by counting the number of times the user found the target chair. The number of successful search for the bi-gram was 30 out of 90 (33.3%), 4-gram was 31 out of 90 (33.3%), and 6-gram was 39 out of 90 (43%). The average number of queries for the bi-gram was 6.28, for the 4-gram was 5.21, and for the 6-gram was 5.03. We demonstrated that with 6-gram, we achieved the best results.

We collected user suggestions about improving the dictionary or the VR interface. Therefore we decreased the total number of words, purging meanings that were not used. We reduced the number of terms in the dictionary, added some shortcuts for the experimenter in order to speed up the response time, and we introduced an initial session where we explained all the words showing a graphical example applied to a chair. With this analysis, and the consideration related on Wizard of Oz approach, we validate hypothesis [H2.1] that states that "Optimised lexical readability for textual information obtained from speech interaction can improve the efficiency in describing an object for retrieval."

4.7 Conclusion

Voice is a powerful way to convey information. User experience benefits from the introduction of the vocal interface used individually or paired with other inputs. A sketch can depict the appearance of an object quickly, both in two dimensions and in three dimensions. In chapter 3, we outlined how, in some situations, sketch interaction is insufficient to achieve high accuracy. In section 4.3.1, we detailed that if the search is performed on a fully variational database, the sketch interaction system encounters difficulties in understanding which colour is associated to a specific part of the model. Therefore, we design a multimodal system to introduce speech to improve the retrieval task for such a database. Introducing speech interaction can help to solve the problem as it increases the search power without affecting the number of user controls or altering the existing functionalities of the application.

In our study, we compare, in a 3D immersive environment, sketch interaction, speech interaction, and hybrid interaction in the context of model retrieval. We defined a dictionary and associated a feature vector to each chair for the speech query. We ran two studies: the first to understand what is the most efficient number of words to consider for a speech query during a search of a target chair. The second study compares the three different modalities in the same searching task. With our solution, we enhance the user search with two independent modes that explore shape and colour feature spaces. We analysed the different strategies created by the user using the hybrid interaction. Our results demonstrate that the hybrid system with sketch and voice improves the outcomes of both the individual interactions, overcoming the problems showed by the sketching system and refining the results of pure voice interaction.

These interactions are supported by ML algorithms that extract features from images or text and search for targets that minimise the differences with user inputs. On the other hand, ML is used in several parts of the visual content production pipeline, including visual metrics that can provide, for example, a scoring method of the rendered image quality. The next chapter will describe how a deep learning model that shares the same nature as the ones used in our interaction studies can be trained to detect and localise artefacts in computer graphics images.

Chapter 5

Dataset and Metrics for Predicting Visible Differences

In the previous chapters, we proposed different strategies within a virtual environment to retrieve objects from sketches using deep learning algorithms. Sketches are studied as advanced forms of interaction finalised to depict a target object. We showed that properly trained deep neural networks could facilitate useful interactions between user and system. The context described in the following sections is part of the visual content production pipeline introduced in Chapter 1. Within the visual pipeline, as part of visual content creation, we focus on visual perception, which is used to assess the quality of displayed images. We implemented deep neural networks to define a new data-driven visual metric, to evaluate the presence of artefacts in images and provide their exact pixel-wise location. This chapter describes how we created the extensive collection of data used for training, the technique used for creating an improved version of the state-of-the-art metrics, and the architecture of our novel deep learning model. In addition, we depict how we designed the statistical model that is part of the loss function during the training process. This study on visual perception was published in [195] and presented at SIGGRAPH 2018.

My personal contribution is threefold. First, helping in writing part of the software (web application) that is used for the experiment. Second, the contribution to implementing the CNN model in Tensorflow. In particular, I implemented the fully convolutional architecture and the statistical loss function. Third, testing our model to produce the results for the super-resolution from the downsampled images application. I didn't contribute to the generation of and the gathering of images and gathering marking data from the user test, training the state-of-the-art metrics, modelling the statistical function, or to the other tested applications (visually lossless compression and content-adaptive watermarking).

5.1 Introduction

Detecting differences in two images is a capability that everybody tested in is own life. Predicting in which part of an image a difference is noticed is a challenging task that visibility metrics try to tackle [35, 152]. However, the achievements are modest, and only straightforward stimuli can be solved effectively such as luminance and contrast masking. On the other hand, for effects correlated with image content and high-level details, the current state of the art is not effective.

We create an extensive collection of images with locally marked distortions to train a predictor of visible artefacts. In total, the images in the database are 557 pairs (distorted image and reference). A subset of the collection (296) is marked manually by 15 or 20 users, while 261 are taken from TID2013 datasets. This collection exceeds the previous largest dataset [151] that includes 37 marked images. In addition, our novel collection consists of different levels of distortion magnitude. We improved the state of the art visible difference metrics training them with our collection and then created a CNN-based metric that outperforms the accuracy of the aforementioned metrics.

Multiple applications can benefit from such a metric prediction. The first application is the reduction of the image size during the compression algorithm in order to achieve a visible lossless result. The second example shows that the metric is capable of identifying the maximum downsampled image to reconstruct a super-resolution image that is considered visually identical to the original. We demonstrate, in a third application, how to achieve a watermarked image that can be considered invisible. In this study, we present different contributions. Although I helped in writing the software for the user test and testing the model for the super-resolution application, my main contribution is the implementation of the deep learning model that predicts artefacts in computer graphics images. Firstly, we generated the most extensive publicly available collection of visible distortions[1] that are manually marked. Secondly, we design a model based on Bayesian inference that determines the probability of detecting a visible distortion in an image. Thirdly, we improved state-of-the-art metrics retraining with such data. Finally, we implemented a visibility metric model based on CNN that outperforms the metrics mentioned above. Besides, we tested our metric in three different applications: visually lossless compression, super-resolution, and invisible watermarking.

5.2 Dataset of visible distortions

My collaborators collect visible distortions in specific image locations. In this way, it was possible identifying which distortions are below the visibility threshold and so not detected, and the visible ones.

[1]The dataset is available at `https://doi.org/10.17863/CAM.21484`

Subset name	Scenes	Images	Distortion levels	Level generation method	Res. [px]	Source
MIXED	20	59	2-3	blending	800×600	custom software, photographs [151]
PERCEPTIONPATTERNS	12	34	1,3	blending	800×800	MATLAB [155]
ALIASING	14	22	1-3	varying sample number	800×600	Unity, CryEngine [196]
PETERPANNING	10	10	1	n/a	800×600	Unity, CryEngine [196]
SHADOWACNE	9	9	1	n/a	800×600	Unity, CryEngine [196]
DOWNSAMPLING	9	27	3	varying shadow map resolution	800×600	Custom OpenGL app [196]
ZFIGHTING	10	10	1	n/a	800×600	Unity, CryEngine [196]
COMPRESSION	25	71	2-3	varying bit-rates	512×512	crops from photographs
DEGHOSTING	12	12	1	n/a	900×900	photographs [197]
IBR	18	36	1,3	varying distance between key frames	960×720	custom software [198]
CGIBR	6	6	1	n/a	960×720	custom software [198]
TID2013	25	261	n/a	n/a	512×384	Kodak image dataset [140]

Table 5.1: Dataset details.

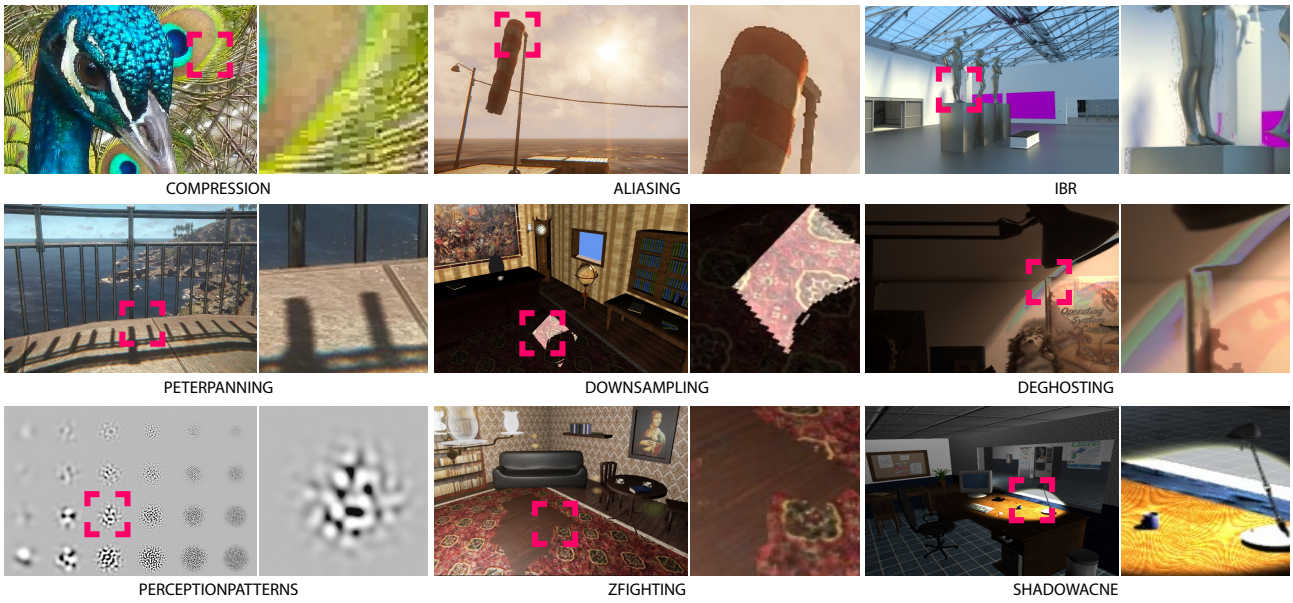


Figure 5.1: The figure shows different samples of stimuli included in our collection. The red square indicates the lateral magnified area where the artefact is located.

In general, the experiments that collect visible distortions uses constant stimuli or adaptive techniques. These operations are quite inefficient, and it can require hours for collecting a limited amount of images (30) such as in [153]. Improving the method described by Čadík et al. in [151], allowed to achieve a largest set. Moreover, images from TID2013 that includes automatically generated marking, were incorporated.

5.2.1 Stimuli

The collection includes 557 images with 170 unique scenes. A large number of them were generated with three levels of magnitude for the specific distortion. The types of the artefacts present in the scenes

are the most common in computer graphics such as image compression, noise, peter-panning, shadow acne, deghosting linked to HDR merging, and warping artefacts generated from image-based rendering technique. Current visibility metrics find this data collection very challenging for the assortment of artefact types mentioned above.

The dataset is constructed from images from preceding researches. Such images are classified in different sets (see Table 5.1 and Figure 5.1). MIXED (59 images) is an extension of the dataset from [151]. These artefacts are virtual point light (VPL), structured and high-frequency noise, light leaking artefacts, clamping, brightness local changes, tone mapping, and aliasing distortions. PERCEPTIONPATTERNS(34 images) from [155] includes artefacts coming from perceptual phenomena. Distortions are related to contrast sensitivity, and luminance and contrast masking. From dataset [196] we extract images (created with game engine like Unreal Engine 4, Unity) with artefacts generated in real-time such as aliasing (ALIASING (22 images)), shadow acne (SHADOWACNE (9 images)), peter-panning (PETERPANNING (10 images)), downsampling (DOWNSAMPLING (27 images)), and z-fighting (ZFIGHTING (10 images)). Near-threshold distortions are included in compression images, COMPRESSION (71 images). Deghosting is generated when high dynamic range (HDR) merging images happens (DEGHOSTING (12 images)) Image base rendering and view interpolation are responsible of the artefacts in the sets named IBR (36 images) and CGIBR (6 images) both extracted from [198]. This collection also includes the set named TID2013 (261 images) from [140] whereas distortions permeated the whole image area or are totally invisible.

5.2.2 Experimental procedure and apparatus

In this section, we describe the experimental setup for marking visible artefacts.

Comparison method Multiple presentation techniques can be used to present a reference image with a distorted one. The flickering method allows a user to switch in place the two images rapidly. On the other hand, a side-by-side presentation places the two images spatially close together with a horizontal layer. A no-reference technique (described in [151]) shows only the distorted image without a term of comparison. The sensitivity to the image differences suffers heavily from the presentation method. Flicker presentation impacts more on observer sensitivity, and it fosters a conservative measure of visible distortions. This attitude affects many applications where, eventually, the reference image is not even visible. Therefore, the side-by-side presentation was selected as more appropriate to computer graphics.

Experiment software We used a browser application that allows users to mark the images with a simple click and drag movement of the mouse. The precision was increased by implementing the

lazy-mouse function. Additional functions such as erase and clear markings were added. Lazy-mouse is a technique that slows down the movement of the brush cursor, increasing the spatial accuracy during the painting task (see the application in Figure 5.2).

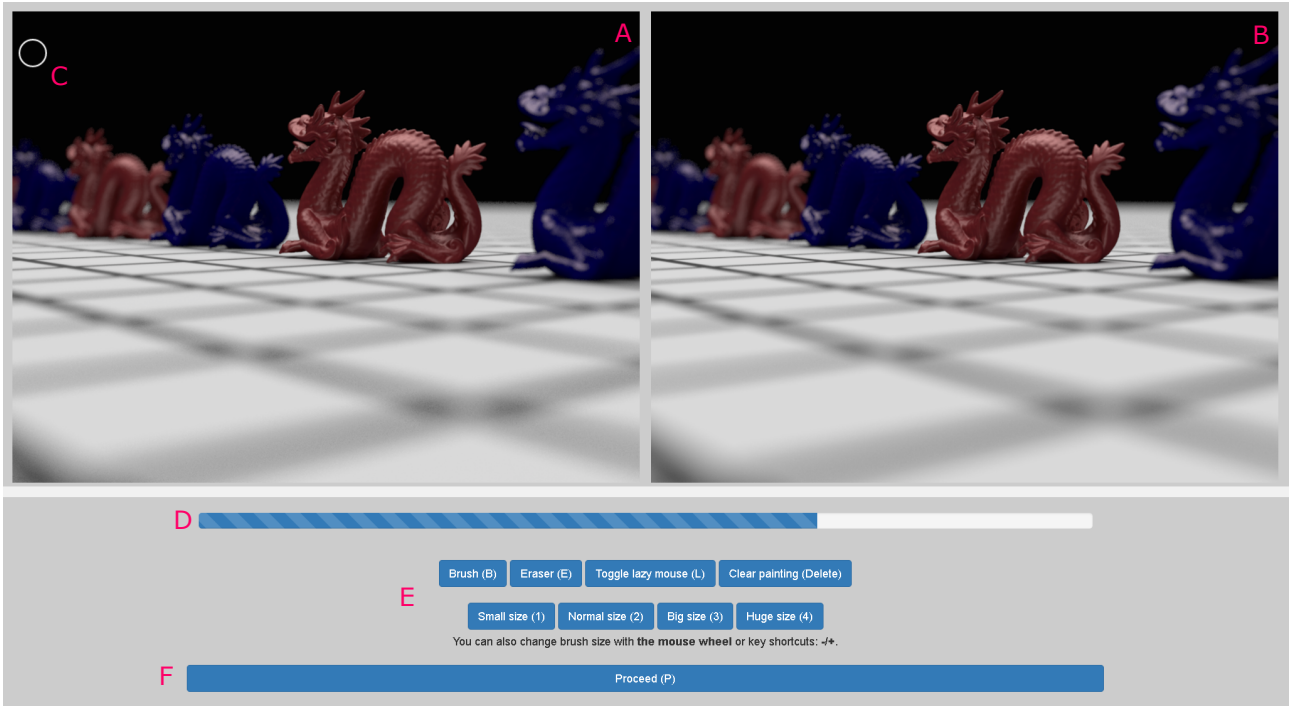


Figure 5.2: View of the browser application used for collecting markings: distorted image (A), reference image (B), brush cursor (C), progress bar (D), settings (E), and continue button (F).

Multiple levels of distortion magnitude Moreover,a mechanism to increase consistency during the marking task was implemented. The distorted images were selected and showed with increasing magnitude of the distortion, up to three levels. The users marked the first level, and the marking achieved from the previous level is inherited and superimposed at the following level (see Figure 5.3). In this way, only new visible distortions receive attention.

Viewing conditions The monitor was positioned to reduce reflections. In addition, the lights were faded in the room. The distance between the observer and the monitor was 60 cm. The monitor was a 23″, 1920×1200 resolution Acer GD235HZ display. Thus, each visual degree had 40 pixels of resolution. The display peak luminance was 110 cd/m^2 and the black level was 0.35 cd/m^2. To scale down artefacts visibility for DEGHOSTING COMPRESSION sets the distance was set in order to have 60 pixels per visual degree.

Observers The participants, recruited from the computer science department, were asked to mark each subset of the entire collection. The total number of observers is 46, aged from 23 to 29, and were paid. None of them had vision-related problems, and they were novel to such an experiment. The experiment

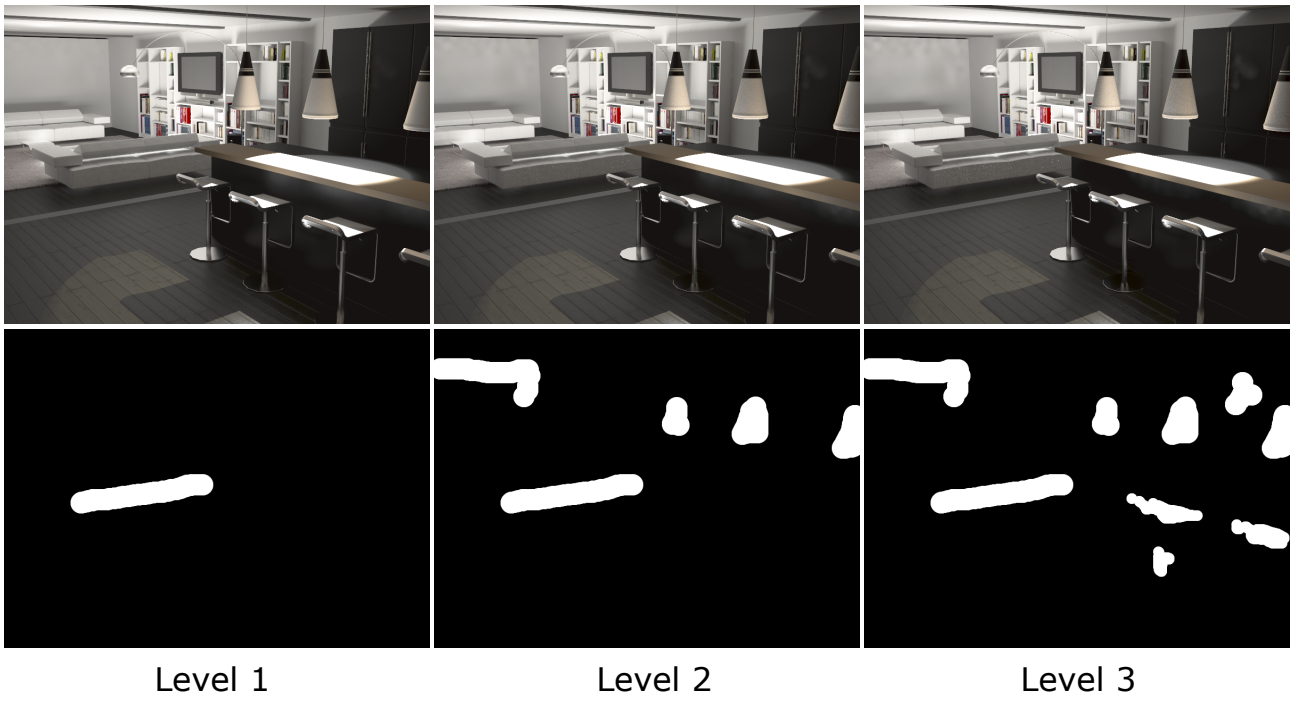


Figure 5.3: In this figure, three levels of increasing magnitude (from left to right) are displayed in the top row and the corresponding marking in the bottom row.

was divided for each user in sessions that last one hour to avoid overexertion (total of 8 hours per user). The final interview showed that the duration of the session was adequate for preventing fatigue.

5.3 Modelling experimental data

Data coming from this experiment is subjected to variance. Therefore the marking results can not be considered as ground truth. Each observer and even the same observer in distinct situations can detect differently, and this impacts the visibility threshold. In addition, if the artefact is not spatially extended in the image, the user needs to search it. This searching component and other multiple causes influence the results. To prevent training the metrics with data subjected to searching component, it is necessary to define a model that deals with this stochastic process in order to cancel effects that distort the prediction. This model is then used inside the loss function during the training process.

In general, visibility metrics try to predict the detection threshold for the average observer from a set of observers. This procedure does not take into account the variance of the distribution for the observers. A different approach instead considers the population instead of the average observer, predicting the fraction of the population that marks the diversity between reference and distorted image. Thus, each scene is assigned from 15 to 20 observers.

The loss function is implemented as the likelihood of observing a difference, given that the probability of detection (p_{det}) is already measured. As depicting in Figure 5.4 in the beginning, it is

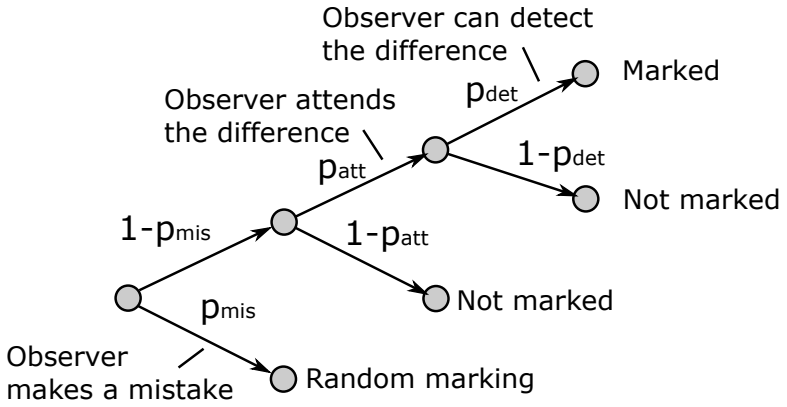


Figure 5.4: This graph defines the statistical model used for the experimental data. The root node shows the probability of making a mistake (p_{mis}) by the observer, then there is the probability of attending (p_{att}) and at the end, detecting (p_{det}) the difference in the image.

possible that the observer made a mistake, painting some area of the image where there was not a difference. This probability of lack of accuracy as p_{mis} (probability of mistake) is equal to a constant value of 0.01 (1% of the cases). The effect of this probability is to avoid a heavy penalisation if the observer is an outlier. The second step relies on understanding why some visible distortions are not marked. The reason is that the user is not attending the location where the distortion is present. For this reason, this model takes into account both the probability of attending a specific location and the probability of detecting the difference. The probability of attending a location of the image is described by p_{att} while p_{det} depicts the likelihood of detecting the diversity. Given a population of N observers where k mark a location in an image, the process can be described as a Bernoulli distribution with the additional mistake probability.

$$\begin{aligned} P(data) &= p_{mis} + (1 - p_{mis}) \binom{N}{k} (p_{att} \cdot p_{det})^k (1 - p_{att} \cdot p_{det})^{n-k} \\ &= p_{mis} + (1 - p_{mis}) \, Binomial(k, N, p_{att} \cdot p_{det}) \, . \end{aligned} \tag{5.1}$$

Visible metrics are focused on predicting the p_{det}. Hence if the difference is visible, expecting that the observer considers all the areas in the image. However to calculate p_{det} it is necessary to know p_{att}. Moreover, an additional challenge is that p_{att} is a variable dependent on observers, artefact types, scenes. In this case, Formula 5.1 describes the probability that if the user does not make a mistake (with probability 1-p_{mis}), the probability of detecting an artifact (p_{det}) conditioned by attending the right area (p_{att}) is modelled by a binomial distribution.

The estimation of p_{att} distribution can be evaluated. The artefact type impacts largely more than the other factors as mentioned earlier. Thus artefacts were classified, grouping them into subsets with

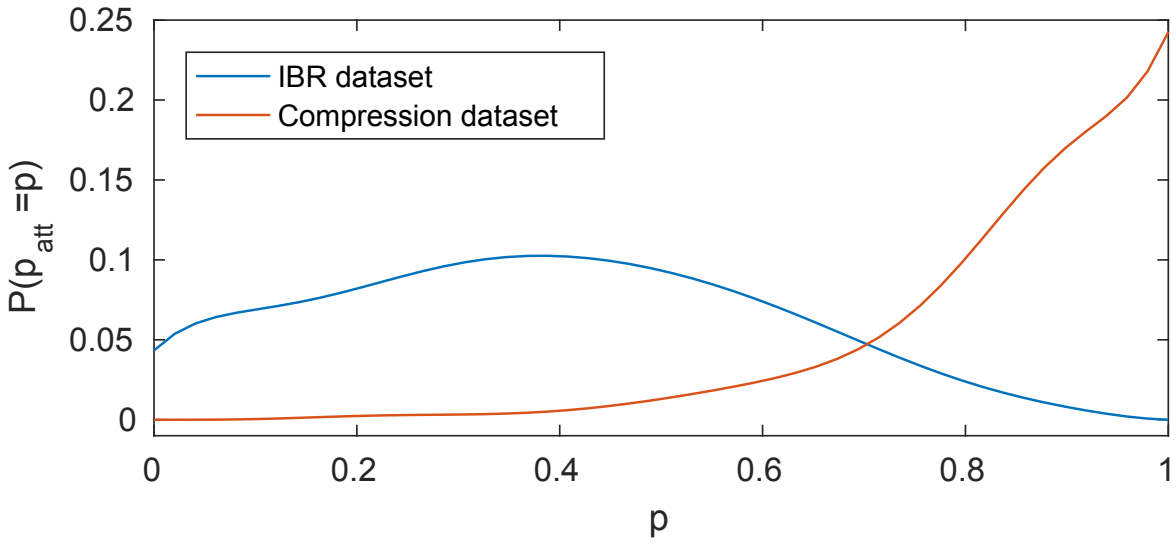


Figure 5.5: The likelihood that the probability of attending the area with the distortion is equal to p, showed for two distinct collections.

similar characteristics, and calculate p_{att} separately. Another important step is a formal definition of visible difference. After having approximated that a pixel is visually detectable if the difference between the values from distorted and reference images is larger or equal to 20 (over 255), given $p_{det} = 1$, p_{att}, the distribution is:

$$P(p_{att} = p) = p_{att}(p) = \frac{1}{|\Omega|} \sum_{(x,y)\in\Omega} Binomial(k(x,y), N, p) \ , \tag{5.2}$$

where Ω is the pixels set with large difference values and $|\Omega|$ its cardinality. For the moment, p_{mis} is not taken in consideration for simplicity. Figure 5.5 shows the distribution of p_{att} of two datasets. The following equation 5.3 measures the logarithm of the likelihood of having p_{det} values estimated by a model that describes the data coming from the experiment:

$$\begin{aligned} L = \sum_{(x,y)\in\Theta} & \log[p_{mis} + (1 - p_{mis}) \\ & \cdot \int_0^1 p_{att}(p) \cdot Binomial\left(k(x,y), N, p_{att}(p) \cdot p_{det}(x,y)\right) dp] \ , \end{aligned} \tag{5.3}$$

with Θ that represents the pixels with x and y coordinates. The second part of the equation is the expectation value when we observe the results having that p_{att} statistics.

When calculating the expected likelihood for two subsets, as shown in Figure 5.6, the probability of having a value of p_{det} depends on how many observers (k) have painted that pixel. In IBR collection

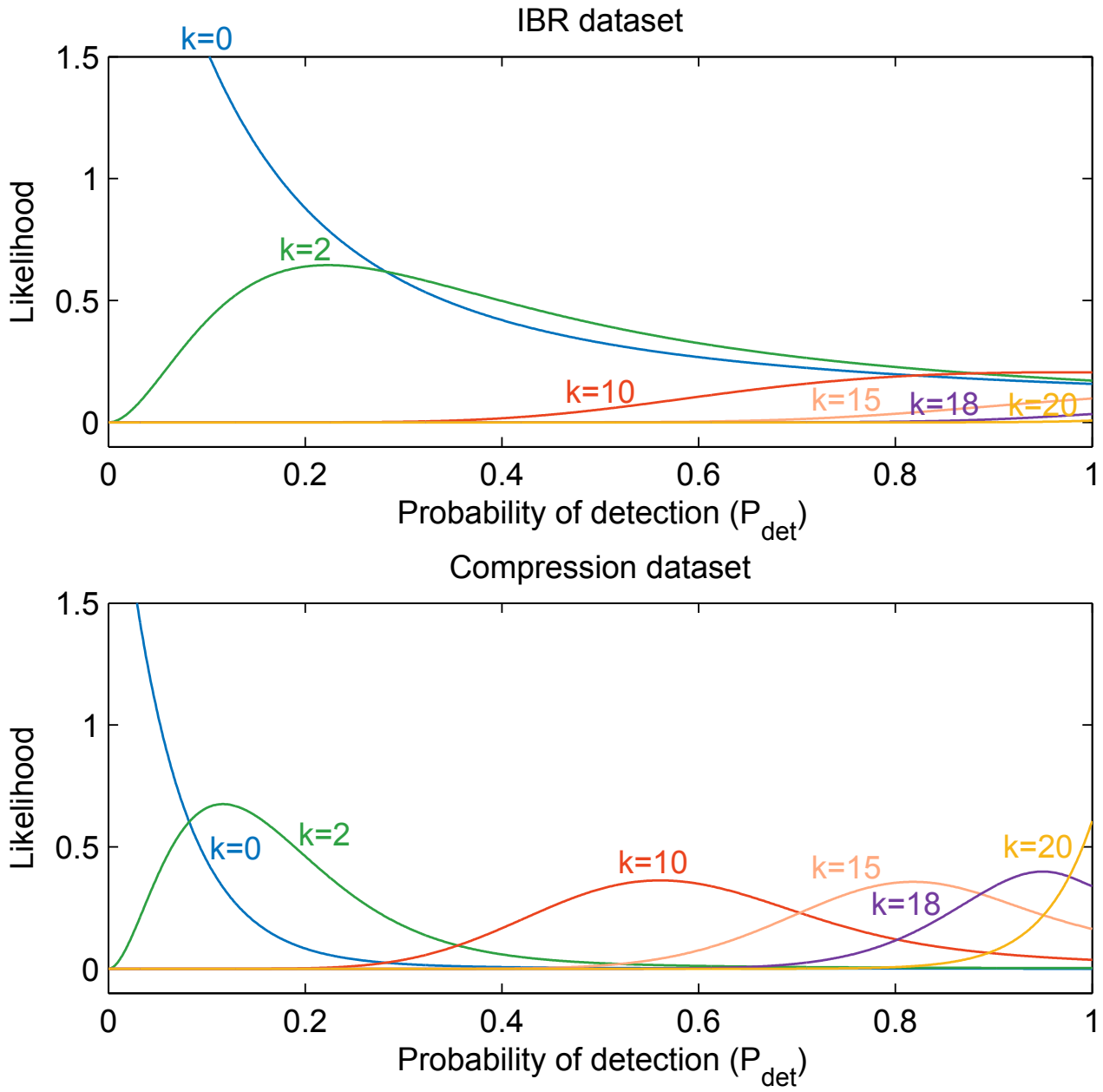


Figure 5.6: Difference detection probability for two datasets.

(top chart) if $k = 10$ observers mark the pixel, p_{det} goes between 0.5 to 1. Therefore, even if only 10 out of 20 observers painted the artefact, it is visible as the difference between images. When the probability of detection corresponding to $k = 10$ observers is around 0.55 (like in compression dataset, bottom chart), it means that 45% of the observers are not able to detect the artefact even if they are looking the area where the distortion is localised.

Noisy data was not directly used, but though Equation 5.3 the probability of detection is estimated considering the distribution of such probabilities for the different datasets.

5.4 Visibility metrics

The most promising quality metrics to produce visible distortions' predictions are selected, and they were trained using the collection. To train such metrics, an adapted version of OpenTuner[2] optimisation software is used, enabled to work in parallel on a cluster dealing with time-consuming calibrations.

[ABS] is the absolute difference (D) between pixels of distorted and reference images. This difference is computed as the weighted sum of RGB gamma-corrected values (luma). After being divided by the threshold (t) it is part of the formula of the psychometric function:

$$p_{det}(x,y) = 1 - \exp\left(\log(0.5)\cdot\left(\frac{D(x,y)}{t}\right)^{\beta}\right) , \tag{5.4}$$

where the pixel coordinates are x and y, and t and β are the optimised parameters. $p_{det}(x,y)$ is inserted in Equation 5.3 to evaluate the loss.

[SSIM] SSIM measures the structure similarities within images providing a number in its final analysis. We transform that result with the following equation:

$$D_{SSIM}(x,y) = \frac{1}{\varepsilon}\left(\log\left(1 - M_{SSIM}(x,y) + \exp(-\varepsilon)\right) + \varepsilon\right) , \tag{5.5}$$

where M_{SSIM} is the value produced by SSIM and $\varepsilon = 10$. D_{SSIM} results in a positive number between 0 and 1 where 0 happens where images are identical. Finally we passed D_{SSIM} to the psychometric function (Equation 5.4) where t, β are still the optimised parameters.

[VSI, FSIM] The same Equation 5.4 is used after transforming the metrics D_{VSI} and D_{FSIM} achieving the same range (0-1) of values. The optimised parameters were still t and β.

[CIEDE2000] The images were transformed in linear XYZ space using gain-gamma-offset from our experimental display. Like the other cases, we evaluated ΔE as probabilities and used in our psychometric function. The fitted parameters were still t and β.

[sCIELab] The same transformation used for CIEDE2000 and used in the psychometric function, is adopted.

[Butteraugli] This metric is based on a constant "good_quality" that does not provide a good result with human test outcomes. Therefore we transformed the visibility map with the psychophysical function in Equation 5.4.

[2] `http://opentuner.org`

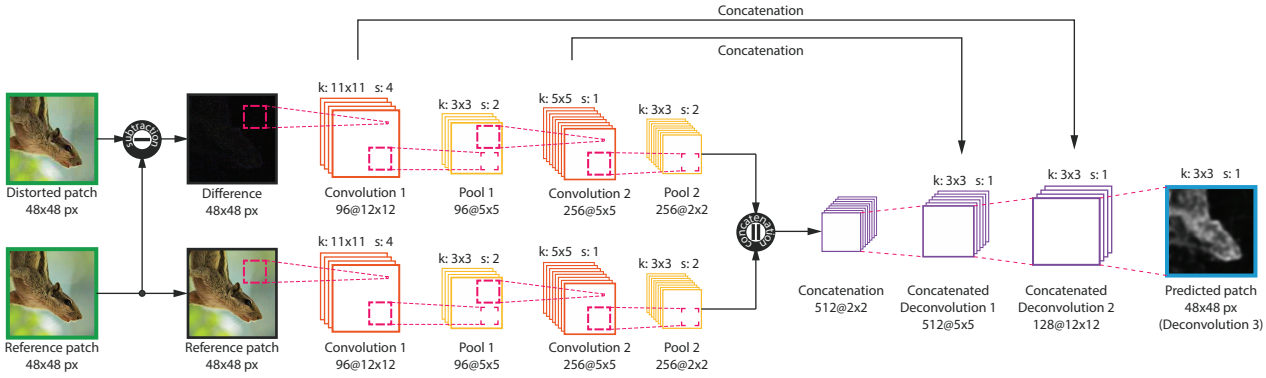


Figure 5.7: Fully convolutional CNN architecture with two branches: the first branch process difference between the reference and distorted image, the second branch takes the reference image. This model produces a visibility map of the same size of input images. Both branches contain two convolution layers where the first layer has 11×11 kernel, stride 4, the second layer with 5×5 kernel, stride 1. The deconvolution part has convolution layers with 3×3 kernel, stride 1.

[HDR-VDP] HDR-VDP v2.2 is modified to improve the predictions. On the one hand, we reduced the multi-scale decomposition by removing orientation-selective bands. On the other hand, we enhanced spatial probability pooling using spatial probability summation to deal with distortions that change in magnitude on the image.

$$P_{sp}(x,y) = 1 - \exp\left(\log(1 - P(x,y) + \varepsilon) * g_\sigma\right)(x,y)\ , \tag{5.6}$$

where $P(x,y)$ is the probability of visibility map (Equation 20 in [35]). While ε is a constant to avoid zero, g_σ is the Gaussian kernel.

All the metrics improved by the training mechanism are characterised with a prefix "T-", e. g., T-SSIM, to make a distinction between the original version and the trained version.

5.5 CNN-based metric

The recent achievements of CNNs in many applications motivated us to design a novel visibility metric with a fully convolutional architecture. Thus, we used our dataset to train it.

5.5.1 Two-branch fully convoluted architecture

Siamese networks became popular in tasks where images comparison is required. Siamese CNN contains two branches that share weights but are fed with different inputs and produce different outputs. The classic Siamese architecture with fully connected layers in the final part was tested. The results were that the performance improved when the difference between distorted and reference component is taken in image space instead of feature space (Bosse et al. [199]) and when the architecture is

fully convolutional. Having two different inputs and avoiding sharing weights improved further the performance.

The used architecture is shown in Figure 5.7. The first branch encodes the difference between distorted and reference images (subdivided in patches) while the second branch accepts reference images (subdivided in patches). In addition, we avoid to share weights between the branches, and we perform a final concatenation of the features maps to preserve them.

We implemented three deconvolution layers that produce the visibility map, and we use skip-connection mechanism. With convolutional layers instead of fully-connected layers, we reduced the number of parameters, preventing overfitting and improving its ability to generalise.

Given R as the reference patch and D as the distorted patch, we have two mapping functions $F_{w^d_{conv}}$ and $F_{w^r_{conv}}$, with w^d_{conv} and w^r_{conv} that are the convolutional layers weights for difference and reference branches. Moreover, given $F_{w_{dec}}$ as mapping function with w_{dec} that is the deconvolution layers weights, our metric $M_w(D,R)$ became:

$$M_w(D,R) = F_{w_{dec}}(Concatenate(F_{w^d_{conv}}(D-R), F_{w^r_{conv}}(R))) \tag{5.7}$$

Convolutional layers Although the used dataset contains many images, we do not have sufficient data to perform a full training. Therefore, we chose to fine-tune the feature extraction layers with a pre-trained model: AlexNet [200].

We compared the results generated by five convolutional layers with the ones generated by four, three and two layers, and we discovered similar outcomes. Thus, we removed the last three layers. Each convolution layer uses a rectified linear unit (ReLU) and is followed by a pooling layer. Moreover, the dropout value is 0.5 to prevent overfitting.

Deconvolutional layers We replaced fully connected layers with deconvolution layers. Deconvolution layers take the outcomes of concatenation operation and create the visibility map. Deconvolution is achieved by a series of upsampling followed by a convolution operation. This solution avoids checker-board patterns that are present when deconvolution use transposed convolution. We concatenate feature maps coming from different branches with deconvolution feature maps with the same size.

5.5.2 Training and testing

We trained our network with the goal of minimising the Equation 5.3, our likelihood function.

We split the images into not overlapping 48×48 pixels patches. After testing different sizes, this patch size results in better visibility maps preserving high-frequencies. Also, to avoid overfitting, we

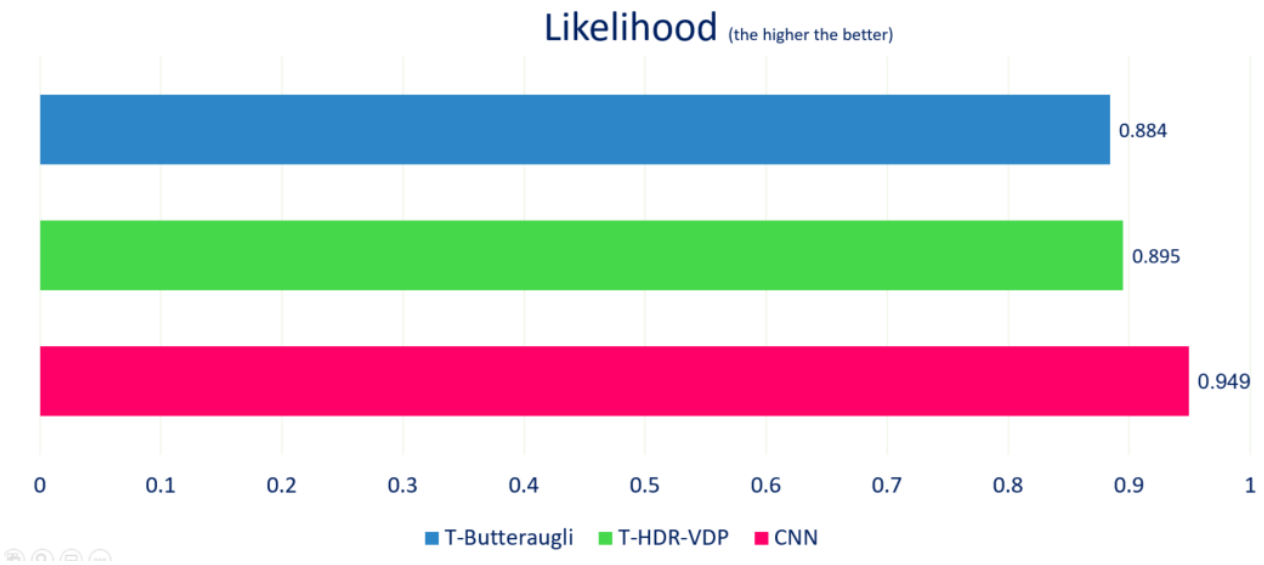


Figure 5.8: This chart shows the average results of the most three significative metrics measured by the Likelihood of detecting an artefact.

increase the number of patches via data augmentation, such as flipping (horizontal and vertical) and rotating 90, 180, and 270 degrees. Before starting the training procedure, we removed all the patches where the difference between reference and the distorted source was null, achieving 400,000 valid patches. Our network was trained with 50,000 iterations using the mini-batch technique (batch size 48). We used Adaptive Moment Estimation (Adam) optimiser with the following parameters: learning rate of 0.00001 and learning decay rate of 0.9.

We implemented the CNN using TensorFlow GPU 1.4 [3]. We trained our network on an NVIDIA GeForce GTX 980 Ti.

Fine-tuned CNN infers the visibility map for each image after a few steps. First, it split the images in 48×48 patches that overlap with 42-pixels. Second, it creates a visibility map for each patch. Third, it reconstructs the full-size visibility map by merging the patched visibility maps and averaging the pixels that belong to overlapping patches. Our CNN predicts the visibility map for a distorted image with 800×600 pixels in 3.5 seconds.

5.6 Metric results

We made a comparison between the predictions of the metrics mentioned before, after a 5-fold cross-validation process. We separate the collection into two subsets with ratio 80:20, where the first was used for training and the second for testing. We generated in this way all the predictions on test images

[3] https://www.tensorflow.org/

avoiding overfitting. The outcomes for the metrics are showed as average in Figure 5.8 and in detail in Figure 5.9, and state that the CNN model outperforms the other metrics despite some datasets are more complex to predict like COMPRESSION and PERCEPTIONPATTERNS.

A second relevant result is that both T-HDR-VDP and T-Butteraugli performed well. T-CIEDE2000 does not perform better than T-ABS, while T-sCIELab performs marginally better than T-ABS.

The transformed and trained versions of the metrics are not oversensitive compared with the original ones (see Figure 5.10) that predicted distortions even when they were invisible.

In Figure 5.11, we can notice the first row where the distorted image of *uncorrelated noise* shows some high-noise circles over a background with lower noise values. While the markings are localised on the circles and close to them, all the metrics predict differences also in areas where only background noise is present. CNN and Butteraugli give better predictions, partly ignoring the differences between reference and distorted images.

Compression is responsible for the artefacts present in *gorilla* image. However, distortion visibility is quite attenuated by the high frequencies of the image, masking de facto the artefacts. The most accurate results come from metrics like CNN, T-HDR-VDP, and T-Butteraugli, highlighting the most visible distortion on the face and ignoring the other parts. In addition, CNN can predict artefacts in the chest area.

The *peter panning* distortion happens when the shadow cast by a 3D model in the scene is detached from it [196]. While T-HDR-VDP and T-Butteraugli exceed in marking the area affected by the artefact, T-FSIM marks the small differences that other metrics properly reject.

The *car* image has distortions on the car body, and in addition, there is a noise pattern marked by a few participants in the bottom corner. Users are not considering artefacts outside the car because they focus their attention on the car itself. Therefore, in this case, our probabilistic function models correctly the data as the users' markings can not be considered the ground truth.

The differences between the distorted and the reference *classroom* image are due to a slight displacement of the camera position during the rendering. In general, it is quite difficult for the observer to detect the misalignment, but a simple binary comparison will highlight many differences. Between the metrics results, CNN showed a better visibility map. The comparison between our metric based with the trained state-of-the-art metrics confirms hypothesis [H3.2] that asserts that "A fully convoluted model boosts its accuracy in predicting visible artefacts using a training set that includes an extensive collection of annotated distortions in computer graphic." In addition, we compared our CNN

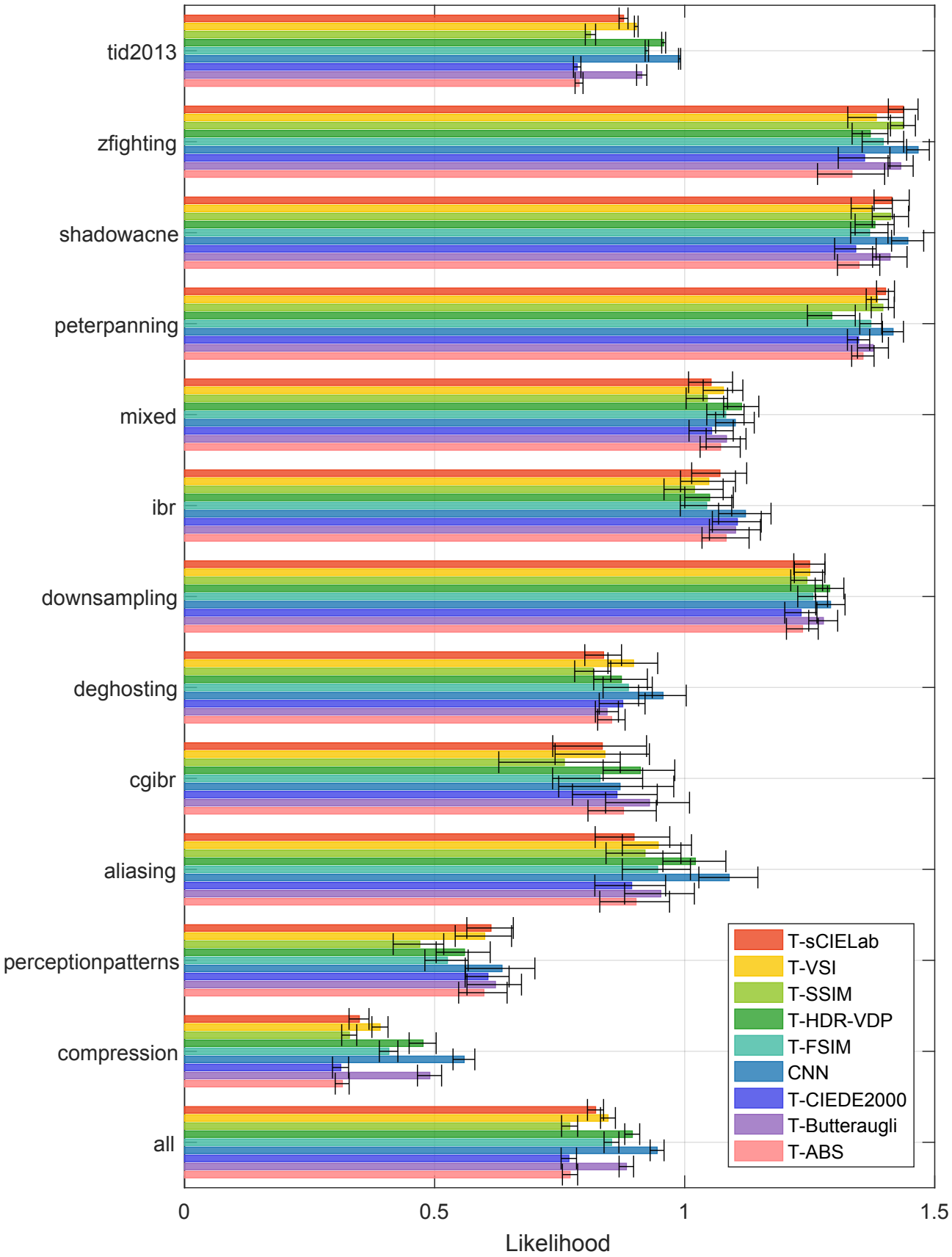


Figure 5.9: All the quality metrics are compared over all the datasets in the collection, using the loss that can measure correctly the probability of detection according to the model. The standard error is showed via the error bars in the chart.

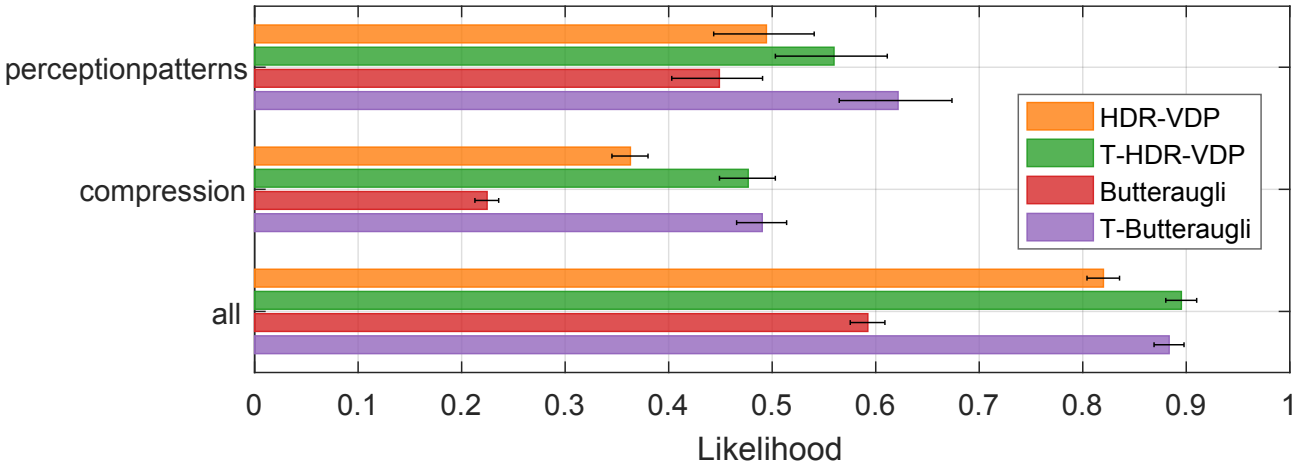


Figure 5.10: This chart shows the improvement of the trained version of two visibility metrics over two datasets.

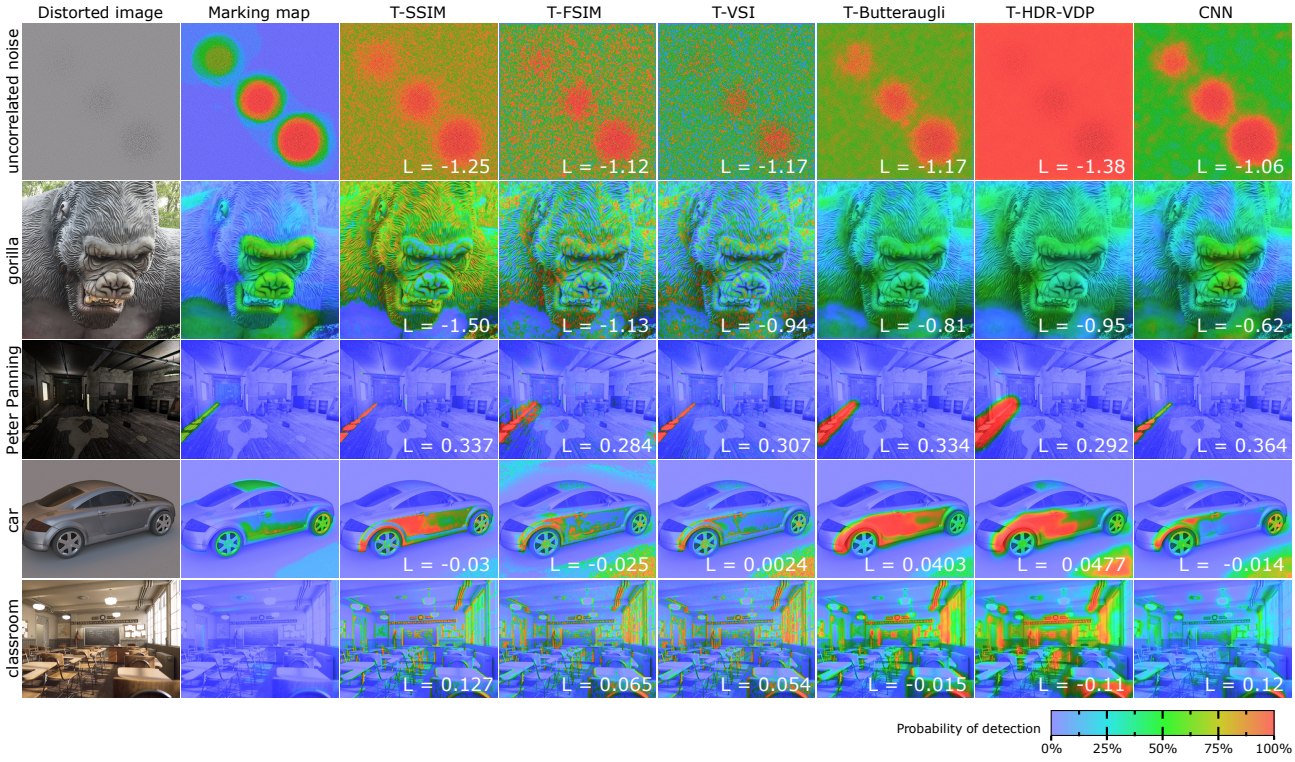


Figure 5.11: From left column we show samples of distorted images, marking maps coming from the user experiment, and the visibility maps from the different metrics.

trained with probabilistic loss, with the same architecture with L2 loss. We proved that with L2 loss, the trained model performed worse as showed in Table 5.2.

	Pearson correlation	Spearman correlation	RMSE
L2-trained model	0.687	0.598	0.167
Probabilistic-Loss-trained model	**0.92**	**0.755**	**0.145**

Table 5.2: L2 loss trained model performs worse if compared with the same architecture trained with probabilistic loss, according to Pearson correlation coefficient, Spearman correlation coefficient and RMSE.

Our approach validates hypothesis [H3.1] that states that "a statistical loss function applied to a CNN-based model improves the accuracy in visible differences detection task".

5.7 Application

In this section we describe one of the three applications implemented in such study. In subsection 5.7.1, the maximum downsampling value for a super-resolution reconstruction algorithm from a single image is evaluated. The other two applications (image compression, watermarking) are described in the paper [195].

5.7.1 Super-resolution from downsampled images

Super-resolution (SR) can generate a higher resolution image starting from a low-resolution image or images. This section describes how we can reduce image resolution from a reference image to achieve a visually indistinguishable full resolution image after applying a single-image super-resolution algorithm.

Three scenes were rendered by using Arnold renderer[4] (Figure 5.12). All the rendered scene present one object on a quasi-uniform grey background. A simple scene was chosen to help the user detecting the distortions.

We select projection onto convex set (POCS [201]) as super-resolution algorithm. We generate SR images from unique LR images. MATLAB was used to produce LR images, and we downsampled the original image by a factor with a range from 1.1 to 6.0.

Our user test was performed using 4AFC QUEST procedure as we did in Section **??**. We gathered distortion visibility data, comparing SR reconstruction with full resolution reference. We calculated the threshold of artefact detection considering downsampling factors. 20 users performed our experiment.

[4] `https://www.solidangle.com/arnold/`

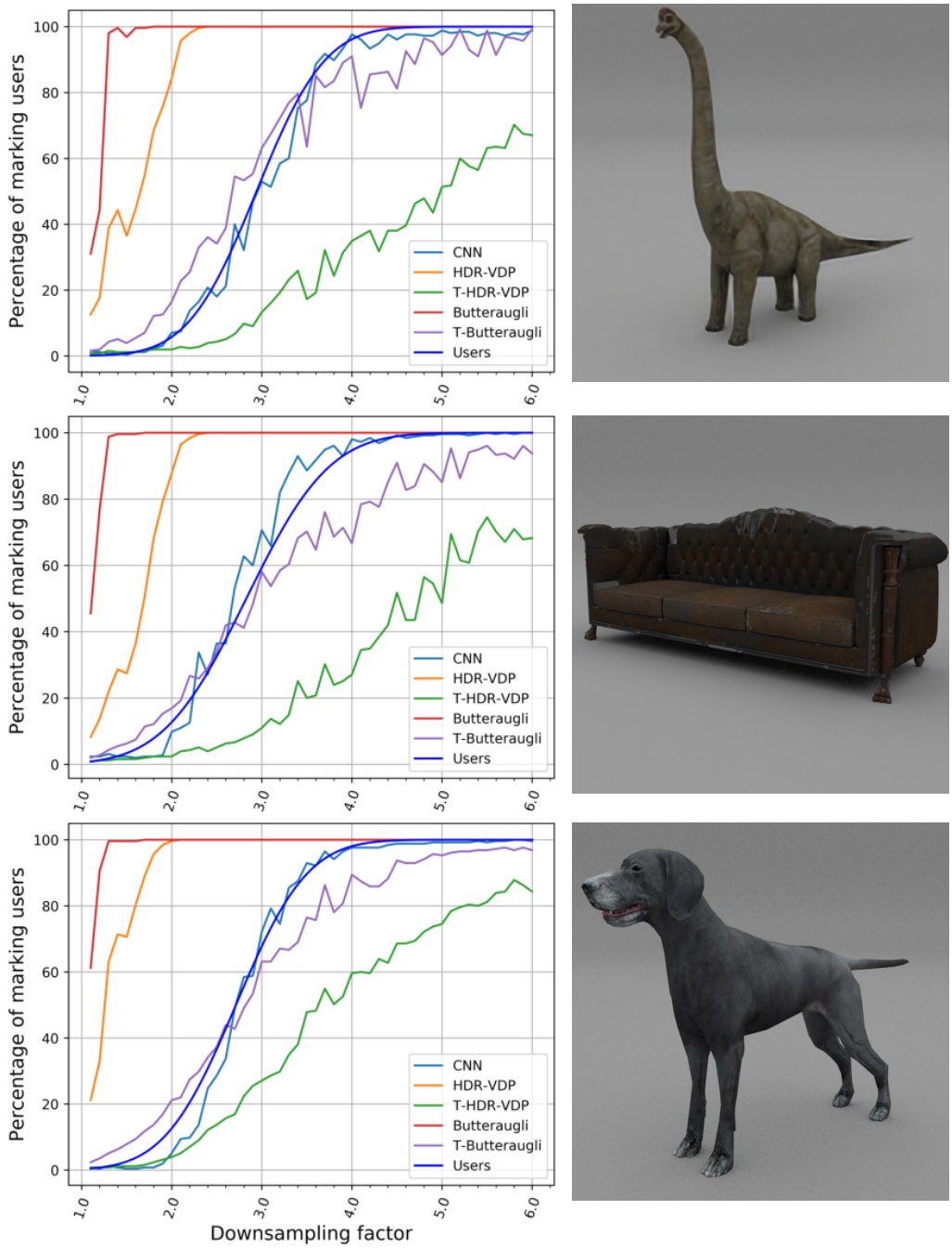


Figure 5.12: We compare our visibility metric results in the super-resolution application. We tested our metric with three scenes (right column) collecting artefact visibility from a user test (blue line in the graph on the left). The chart shows the metric predictions with different downsampling factors.

Figure 5.12 displays with the blue line the population percentage that chose the lower downsampling than the one present on the x-axis. This line is smooth because it is the cumulative plot after fitting a Gaussian distribution to the collected thresholds. Metric predictions are calculated by taking the maximum value of the visibility map (maximum visible distortion). We compared metric performance by calculating MSE between user data and each prediction for all the downsampling

factors. Therefore we average MSE value for all the scenes. CNN scored an error of 19.95, resulting in the most accurate metric. T-Butteraugli followed with an error of 96.96.

5.8 Limitations

Our data collection, despite its efficiency, is not free from problems. Firstly, find and detect an artefact are not completely separated. We adjust our loss function in order to address this issue. Anyway decreasing the image size can help to reduce the searching component in particular for COMPRESSION. Secondly, the experiment procedure requires time for gathering a collection reliable for being used as input for a neural network. In addition to the moment, our system does not take in account possible variations of the distance between the observer and the display, neither change in luminance.

5.9 Conclusions

Visible artefacts in images are difficult to predict correctly. The state-of-the-art metric still lacks robustness and do not work for all the kind of distortions. The challenging aspects are strictly related to the complexity of the human visual system. Our work tries to fill the gap between computational metrics and visual system. We gather an extensive collection of data and training models over it. In addition, as different components contribute to the perception of one or more differences between images, we developed a statistical system that tries to exclude the searching component during training systematically.

We demonstrate that a data-driven metric is able to perform better than the untrained counterpart. In addition, we introduce a CNN model that outperforms the other metrics. These metrics can be applied to multiple applications where a quality need to be evaluated or where a parameter is tuned with the possibility of avoiding loss of quality.

Chapter 6

Conclusion & Future Work

This chapter outlines the contributions of the thesis, respect to the research hypotheses delineated in chapter 1.

6.1 Summary of Contributions

This thesis explored advanced interactions that use textures (and fine details) for searching objects and making a prediction in visual difference detection. The sketch is a simplified drawing that can convey both an abstract idea and also a specific instance of an object in a category. Sketching is one of the most promising and intuitive interactions that has been brought to computer graphics in the last decades. We have shown that in an immersive environment, converting a 3D sketch into a series of textures produces an efficient input to search among a collection of objects (ShapeNet - chairs). In this way, we contribute to the field of Computer Vision, Information Retrieval, and Human-Computer Interface. We have also shown that a combination of sketch and speech can improve further the accuracy of the search. Image quality metrics measure the difference between two images, assigning a score of similarity. On the other hand, visual metrics are able to localise the difference in producing difference maps and are used to detect distortions. In the context of visual perception, we demonstrate that using textures as the training set for visible differences as input to a CNN-based statistical model outperforms the state of the art of visible metrics. In this way, we contribute to the field of Computer Graphics (more specifically in the sub-field of Quality Assessment).

In Chapter 1, we stated research hypotheses pertinent to applications both related to images used in the context of sketch-based retrieval and visible differences predictions. In the next sections, we illustrate the contributions achieved in each chapter, showing how they support such hypotheses.

6.1.1 3D Sketching for Interactive Model Retrieval in Virtual Reality

Chapter 3 introduced a novel system for retrieving items from a 3D model collection with fine details using 3D sketch as interaction in virtual reality. We are all trained from a young age to draw on surfaces usually flat, and 3d pens only recently exploit the depth component in real-world [202]. However, 3D sketch in a virtual environment became possible with the advent of position trackers, and only in the last few years gained interest, mainly for artistic purposes, thanks to the commercialisation of cheap virtual reality devices. This trend legitimises the question of how 3D sketch can be exploited as a new form of input beyond the artistic intent, exploring its efficiency and feasibility. Previous works on information retrieval systems based on 3D sketch are oriented to retrieve objects that belong to different categories avoiding collections of a unique type of item with fine details. In addition, previous works did not achieve satisfactory results due to the representative nature of the sketch that is not converted into a proper descriptor. Our system makes use of the CNN-generated descriptor from the combination between the sketch and a background model to improve the accuracy of the search. The promising results obtained by our system supports the following hypothesis:

[H1.1]3D sketch is an effective interaction to depict information in an immersive environment with the aim to retrieve models from a large collection.

Proposing this way of communication in a virtual environment opens the doors to additional possibilities such as adding functionalities such as undoing instantaneously or erasing parts of the sketch. In this context, free-hand strokes can also be extended to the third dimension as sketch visibility can be fixed in the space as the pencil stroke is fixed over a sheet. We tested different modalities of sketching to determine the most effective for our searching task: 2D Virtual Tablet, 2D Whiteboard, 2D Real Tablet, and 3D Sketch. We found that the 3D sketch was the most accurate method for retrieval. The 3D sketch is able to depict the different points of view, exploiting the additional dimension. Our system generates 12 snapshots with strategic points of view of the 3D sketch and makes a descriptor comparison with the objects in the database. This methodology is used extensively as interaction input to support all the following hypothesis:

[H1.2] Deep learning models are more effective than well-known detectors and descriptors when applied to sketches in a three-dimensional context.

Algorithms such as SIFT, HOG, GF-HOG and colorSIFT are extensively used to generate image descriptors e. g., for object detection or segmentation. As sketch is a representation of an object instead

of an image of a real one or even synthetic one, it is influenced by different factors: user skills, cultural background, and even mood. We tested with the same textures the accuracy produced by each of the descriptors proposed, comparing them with the features extracted by a convolutional neural network. We defined a set of similarity rules that involves structure, style and colour features to produce a score used to rank the different methods. The final results showed that the CNN model is the most accurate and has been used as the ultimate backend in our sketch-retrieval system.

[H1.3] Feature descriptors generated by the combination of sketch and 3D model in the context of an interactive loop, are an improvement in terms of accuracy.

A Sketch alone, even if detailed and carefully coloured, may not be enough to create an accurate descriptor for the CNN. The group of strokes that makes up the sketch are subjected to inaccuracies such as incorrect strokes, noise, and voids that are not filled by the user. These issues can impact profoundly on the search, and most of them can not be corrected. To improve the quality of the search, we introduced the possibility of mixing up the sketch descriptor with a 3D model descriptor. When a model is selected from a retrieval panel, the user places it in the environment. The user can superimpose the sketch on the model in the textures to depict both gross and fine details simultaneously. This solution showed an improvement in the search that supports this hypothesis.

6.1.2 Multimodal approach fusing sketch and speech in Virtual Environment

In Chapter 4, we illustrated a multi-modal system for object retrieval where sketch and speech are fused together in virtual reality. By generating the multi-view descriptor of sketch and model, it was possible to identify the target object among a large collection quickly. However, ShapeNet is deficient in inter-model variation – the colour and texture of the parts of the objects, and this lack helps the search significantly. We created a chair dataset with intra-object variation and semantic tags called Variational Chair ShapeNet (VCSNET). We found that with this dataset, our system exhibits problems in identifying the exact chair correctly. We analysed the efficiency of speech interaction, and we needed to determine what is the best value of lexical readability to describe the chair. In our case, a simple measure of the length of the sentence was sufficient. We designed a test with experimenter-in-the-loop and measured the results for supporting the following hypothesis:

[H2.1] A optimised lexical readability for textual information obtained from speech interaction can improve the efficiency in describing an object for retrieval.

We tested three options of n-grams to establish the best query length during a pure speech

interaction session. An experimenter-in-the-loop design handles tokenisation and lemmatisation. It helps to achieve very quickly, through a graphic interface, a semantic descriptor with features belonging both to colour and shape information. While a long query is not preferable as in recurrent models, where previous words tend to be forgotten compared to recent ones, we showed that the most robust value for query length, considering the ceiling effect, is six semantically valid words. This setting is subsequently used in the next user experiment that compared the hybrid form of interaction with the individual ones. Sketch and speech queries are built, starting from diverse input channels. This difference implies that two different feature spaces are explored with visual and textual criteria, and the user can jump from one to the other in a natural way. However, they can be managed to follow a similar architecture to be compatible with a query pipeline. We analysed the query type structure supporting the following hypothesis:

[H2.2] A multi-modal system including sketch and speech, to work properly, needs a formal query definition and a specific interaction pipeline.

We formalised both the queries to contains three mandatory components. Firstly, the input component coming directly from user interaction, eventually processed by the experimenter-in-the-loop. Secondly, the descriptor is processed, and the system proposes a chair set to the user. Finally, the user selects a chair from the resulting set, replacing the model in the environment. This final chair represents the input for the next query, regardless if will be a sketch or speech query. In this way, we defined our pipeline as a sequence of any queries set one after the other. A query that is interrupted without selecting an item from the proposed set is rejected automatically. Therefore, we avoid the risk of asynchronous events and contradictory information that can be generated simultaneously. Therefore, we compare the different interactions achieving results that support our hypothesis:

[H2.3] The combination of voice and sketch interaction improves the search in an immersive context when compared to individual techniques.

With our approach, we compared three different methods of search: speech interaction, sketch interaction, hybrid speech-sketch interaction. The results showed that the hybrid method is the most robust method to identify the target chair. Users tended to use speech to define colour components and shapes while they used the sketch to detail shapes as expected. In addition, the speech component was preferred in terms of user experience as more natural and powerful than the 3D sketch. Sketch alone is able only to suggest the correct shape. It encounters considerable difficulties in associating the colour

or texture with the right component of the chair.

6.1.3 Dataset and metrics for predicting visible differences

Chapter 5 presented a novel deep-learning-based visual difference metric between images that make use of an extensive dataset of computer graphics distortions. An extensive dataset of images containing different types of distortion with diverse levels of magnitude is used to train a model with a novel statistical loss function. This database is the most extensive in terms of computer graphic artefacts and contains distortions coming from computer graphics such as peter-panning, shadow acne, down-sampling, z-fighting, compression, deghositing and perceptual patterns. A trainable version of the most diffuse visual metrics is used as a strong baseline for comparing the results. A CNN-based statistical model is created to support our research hypothesis:

[H3.1] A statistical loss function applied to a CNN-based model improves the accuracy in visible differences detection task.

The model is based on a statistical process that keeps in consideration the probability of attending the artefact, the probability of detecting the artefact, and the probability of making a mistake in one of these two stages. The achieved loss function is used to train a two branch CNN-based model that takes as input the reference image and the distorted one. Using classic loss functions such as mean square error does not consider essential aspects of the process of detecting the artefact, such as the separation between the searching component. This component can affect the difference map creation, achieving a magnification of the marked map as a result.

[H3.2] A fully convoluted model boosts its accuracy in predicting visible artefacts using a training set that includes an extensive collection of annotated distortions in computer graphic.

The overfitting issue is managed in two different ways. Firstly, by augmenting the data with rotated and flipped images. Secondly, by reducing the parameters of our model by converting the architecture to a fully convoluted network that makes use of residuals coming from the previous layers. The FR predictor is then compared with the state-of-the-art metrics empowered with a trainable process. The achieved visible maps are more accurate and outperform classic and well-known algorithms that are used to rank similarities between images. This CNN model anyway is not able to predict distortions that are not inserted in the original dataset. Therefore, a fine-tuning procedure is defined for extending the image collection, and the training in case of new artefacts need to be detected.

6.2 Future Work

This thesis has already identified, throughout its contributions chapters, specific high-level indications for future works and research. We summarise them in two different categories that map the main arguments treated in this work; advanced sketch interaction (Section 6.2.1) and visual perception (Section 6.2.3).

6.2.1 Advanced sketch interaction

This thesis showed an advanced sketch-based interaction where the user depicts an item with a set of strokes. He/she confers a 3D shape and colours to it to retrieve the correct object with fine-grained features from a large collection. We decided to restrict the number of possible functionalities: sketching is the predominant interaction, colours can be selected by a palette, undo function is implemented to remove strokes. We did not implement a sketch erasing tool, or a "fill space" functionality. This strategy forced the user to spend most of the time in sketching without exploring the interface to search for shortcuts. It is easy to imagine using gesture recognition for object type identification (table, lamp, etc.), avoiding NLP or text selection. Alternatively, familiar tools from photo editing e. g., brushing aid in depicting large region colour or fill-bucket tool to specify the texture of a region (an element not easily represented). A more sophisticated system equipped with such complementary interactions could be developed for more complex tasks. Indeed, while our implementation allows us to perform controlled user studies about how users sketch within a virtual reality, there are many applications in fields that require real-world interaction. Creative industries are rapidly moving to produce contents for virtual and augmented reality devices. By adaptations to this method, we could see applications within film & TV for set design, game design, or architecture for model creation.

Recent developments in sketch-based interaction show that there is interest in assisted drawing system, such as ShadowDraw [53]. Relevance feedback has been shown to be beneficial [203, 204] as it generates a response for each new stroke and guides the user more quickly to the right query. A possible follow-up of our study is to adjust the current system to provide user-independent real-time feedback that superimposes the best model automatically according to the current version of the sketch. Relevance feedback requires a faster response from the system. The user would not trigger the search or wait for the termination of the sketch. It would allow the user to navigate the database more accurately and shows potential diversion from the road toward the solution.

6.2.2 Sketch and voice interaction

In Section 4.5.1.1, we described different approaches to a possible sketch and vocal pipeline that starts with the verbal description of an object and terminates with the presentation of a subset of potential targets from a database. We selected a semi-automatic pipeline delegating the experimenter the tasks of speaker recognition, speech recognition, NLP, and an automatic process to generate the descriptor and models selection. Our choice was motivated mainly by the low accuracy achieved by some untrained speech recognition services, that tamper the entire pipeline on early stage. This lack of precision is also caused by the quality of the Oculus Rift microphone that impacts the result of dictation software. A possible follow up of our research is the implementation of a fully automatic speech pipeline. Having a real-time and accurate pipeline is a challenging task. Both these features deteriorate rapidly if only one stage of the pipeline underperforms. While latency introduced by one stage accumulates and affects only the final time, loss of accuracy directly impacts the input of the next stage and undermines the entire pipeline. A streaming pipeline version with a temporal window of interest can mitigate a low accuracy issue, excluding that event after a few seconds. Moreover, such streaming pipeline alleviates the real-time issue, achieving results continuously instead of waiting for the sentence's end. However, different stages need to be fine-tuned to achieve a reliable pipeline. For example, speech recognition accuracy can be improved considerably with a training session where the user provides labelled samples of vocal records. In addition, in the environment, only the speaker voice should be present. The audio-to-text conversion stages are tokenisation, text cleaning, lemmatisation, and stemming which are common functions of the NLP libraries. They perform fast and accurate. Also, text classification can be provided by Transformer models (described in 4.5.1.1) which are state of the art at the moment of NLP models.

Our experiment explored the possibility to fuse sketch and speech as descriptive methods while incorporated in the system loop. Despite concerning two different forms of expressions (visual and vocal), we formalised queries as a sequence of analogous steps and uniformed the results. This approach allows us to abstract query objects, process their results independently from their source, and concatenate them in a content creation pipeline. Moreover, this generalisation can be extended to other input forms (such as gestures) depicting a multi-query workflow.

6.2.3 Visual metrics

Visual metrics play a significant role in assessing images to implement rendering algorithms. In this thesis, we explored how our data-driven visual difference predictor improves the current state of the art of visual metrics. Despite the excellent results in terms of accuracy, some aspects of the model can be

improved. Firstly, the method to collect human markings is time-consuming and laborious for the user. Providing a fast way to produce a training set is a contribution that we address in a recent study [205], but it could be further improved. In this case, the training set is generated with a user test that does not require a demanding human marking but an affordable selection between different images. Secondly, the model that splits input in patches to predict local markings of an image is not compatible with real-time requirements. An interesting development would be a real-time version of the model with the possibility of embedding the predictor in a game engine. With a real-time predictor, another attractive idea is to apply such a model in a VR/AR device to detect, for example, the highest downsampling factor that can be applied to a texture that does not produce any perceptible distortion. Finally, another interesting direction for research would be to extend the model to treat also a continuous range of values for display parameters such as peak brightness, or contrast or viewing distance.

Chapter 7

Publications

1. **Model Retrieval by 3D Sketching in Immersive Virtual Reality** Giunchi D., James S., Steed A., 2018 IEEE VR Poster
2. **3D Sketching for Interactive Model Retrieval in Virtual Reality** ,Giunchi D., James S., Steed A., 2018 Expressive Paper
3. **Dataset and metrics for predicting visible differences**, Wolski K., Giunchi D.,Ye N., Didyk P., Myszkowski K., Mantiuk R., Seidel H., Steed A., K. Mantiuk R., 2018 SIGGRAPH Paper
4. **Mixing realities for sketch retrieval in Virtual Reality** Giunchi D., James S., Degraen D., Steed A., 2019 VRCAI Poster
5. **Mixing Modalities of 3D Sketching and Speech for Interactive Model Retrieval in Virtual Reality** Giunchi D., Sztrajman A., James S., Steed A., 2021 IMX Paper

Bibliography

[1] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *CVPR*, pages 1912–1920. IEEE Computer Society, 2015.

[2] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, International Conference on Computer Vision '15, pages 945–953, Washington, DC, USA, 2015. IEEE Computer Society.

[3] Mathias Eitz, Ronald Richter, Tamy Boubekeur, Kristian Hildebrand, and Marc Alexa. Sketch-based shape retrieval. *ACM Trans. Graph.*, 31(4):31:1–31:10, 2012.

[4] Yongxin Yang and Timothy M. Hospedales. Deep neural networks for sketch recognition. *CoRR*, abs/1501.07873, 2015.

[5] Zhou Wang and Alan C. Bovik. *Modern Image Quality Assessment*. Morgan & Claypool Publishers, 2006.

[6] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *British Machine Vision Conference*, 2014.

[7] Gretchen McCulloch. All things linguistic. `https://allthingslinguistic.com/`, May 2014.

[8] Cisco. The Zettabyte EraTrends and Analysis, 2013.

[9] Yu Xiang, Wonhui Kim, Wei Chen, Jingwei Ji, Christopher Bongsoo Choy, Hao Su, Roozbeh Mottaghi, Leonidas J. Guibas, and Silvio Savarese. Objectnet3d: A large scale database for 3d object recognition. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII*, pages 160–176, 2016.

[10] Kaichun Mo, Shilin Zhu, Angel X. Chang, Li Yi, Subarna Tripathi, Leonidas J. Guibas, and Hao Su. PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[11] Unity3D. `https://www.unity.com/`.

[12] Unreal Engine. `https://www.unrealengine.com/`.

[13] Richard A Bolt. put-that-there voice and gesture at the graphics interface. In *Proceedings of the 7th annual conference on Computer graphics and interactive techniques*, pages 262–270, 1980.

[14] Pedro Manuel Antunes Sousa and Manuel J. Fonseca. Sketch-based retrieval of drawings using spatial proximity. *J. Vis. Lang. Comput.*, 21(2):69–80, 2010.

[15] Daniel F Keefe, Daniel Acevedo Feliz, Tomer Moscovich, David H Laidlaw, and Joseph J LaViola Jr. Cavepainting: a fully immersive 3d artistic medium and interactive experience. In *Proceedings of the 2001 symposium on Interactive 3D graphics*, pages 85–93, 2001.

[16] David Bischel, Thomas Stahovich, Eric Peterson, Randall Davis, and Aaron Adler. Combining speech and sketch to interpret unconstrained descriptions of mechanical devices. In *Proceedings of the 21st International Jont Conference on Artifical Intelligence*, IJCAI'09, pages 1401–1406, San Francisco, CA, USA, 2009. Morgan Kaufmann Publishers Inc.

[17] Roman Zenka and Pavel Slavik. Supporting ui design by sketch and speech recognition. In *Proceedings of the 3rd Annual Conference on Task Models and Diagrams*, TAMODIA '04, pages 83–90, New York, NY, USA, 2004. ACM.

[18] Oculus. `https://www.oculus.com/`.

[19] HTC Vive. `https://www.vive.com/`.

[20] Green Light Insights. `https://greenlightinsights.com/`.

[21] Leap Motion. `https://www.leapmotion.com/`.

[22] Kenneth Moreland. A survey of visualization pipelines. *IEEE Trans. Vis. Comput. Graph.*, 19(3):367–378, 2013.

[23] William J Schroeder, Kenneth M Martin, and William E Lorensen. The design and implementation of an object-oriented toolkit for 3d graphics and visualization. In *Proceedings of Seventh Annual IEEE Visualization'96*, pages 93–100. IEEE, 1996.

[24] Zezhou Cheng, Qingxiong Yang, and Bin Sheng. Deep colorization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 415–423, 2015.

[25] George Peter Kenneth Carr, Hoang M Le, and YUE Yisong. Data-driven ghosting using deep imitation learning, June 7 2018. US Patent App. 15/830,710.

[26] Raquel Vidaurre, Dan Casas, Elena Garces, and Jorge Lopez-Moreno. Brdf estimation of complex materials with nested learning. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1347–1356. IEEE, 2019.

[27] Zhenyu Shu, Chengwu Qi, Shiqing Xin, Chao Hu, Li Wang, Yu Zhang, and Ligang Liu. Unsupervised 3d shape segmentation and co-segmentation via deep learning. *Computer Aided Geometric Design*, 43:39–52, 2016.

[28] Daniel Holden, Taku Komura, and Jun Saito. Phase-functioned neural networks for character control. *ACM Transactions on Graphics (TOG)*, 36(4):1–13, 2017.

[29] Patsorn Sangkloy, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays. Scribbler: Controlling deep image synthesis with sketch and color. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5400–5409, 2017.

[30] Junyuan Xie, Linli Xu, and Enhong Chen. Image denoising and inpainting with deep neural networks. In *Advances in neural information processing systems*, pages 341–349, 2012.

[31] DeLiang Wang and Jitong Chen. Supervised speech separation based on deep learning: An overview. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(10):1702–1726, 2018.

[32] Yu Wu, Wei Wu, Chen Xing, Ming Zhou, and Zhoujun Li. Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots. *arXiv preprint arXiv:1612.01627*, 2016.

[33] Pavlo Molchanov, Shalini Gupta, Kihwan Kim, and Jan Kautz. Hand gesture recognition with 3d convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 1–7, 2015.

[34] Kyle Krafka, Aditya Khosla, Petr Kellnhofer, Harini Kannan, Suchendra Bhandarkar, Wojciech Matusik, and Antonio Torralba. Eye tracking for everyone. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2176–2184, 2016.

[35] Rafal Mantiuk, Kil Joong Kim, Allan G. Rempel, and Wolfgang Heidrich. HDR-VDP-2: A calibrated visual metric for visibility and quality predictions in all luminance conditions. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 2011. Article 40.

[36] Guillaume Lavoué. A multiscale metric for 3d mesh visual quality assessment. In *Computer Graphics Forum*, volume 30, pages 1427–1437. Wiley Online Library, 2011.

[37] Jongyoo Kim and Sanghoon Lee. Deep learning of human visual sensitivity in image quality assessment framework. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1676–1684, 2017.

[38] Weilong Hou, Xinbo Gao, Dacheng Tao, and Xuelong Li. Blind image quality assessment via deep learning. *IEEE transactions on neural networks and learning systems*, 26(6):1275–1286, 2014.

[39] Sean Bell and Kavita Bala. Learning visual similarity for product design with convolutional neural networks. *ACM transactions on graphics (TOG)*, 34(4):1–10, 2015.

[40] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, Nov 2004.

[41] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*, pages 886–893, 2005.

[42] Rui Hu and John Collomosse. A performance evaluation of gradient field hog descriptor for sketch based image retrieval. *Comput. Vis. Image Underst.*, 117(7):790–806, July 2013.

[43] Alexa M. Eitz M., Hays J. How do humans sketch objects? In *ACM Trans. Graphics, 31(4)*, 2012.

[44] Shinde J.V. Birari D.R. Survey on sketch based image retrieval. *International Journal of Advanced Research in Computer and Communication Engineering*, 4(12), 2015.

[45] Jonathan Ashley, Myron Flickner, James L. Hafner, Denis Lee, Wayne Niblack, and Dragutin Petkovic. The query by image content (QBIC) system. In *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data, San Jose, California, May 22-25, 1995.*, page 475, 1995.

[46] Alberto Del Bimbo, Pietro Pala, and Simone Santini. Visual image retrieval by elastic deformation of object sketches. In *Proceedings IEEE Symposium on Visual Languages, St. Louis, Missouri, USA, October 4-7, 1994*, pages 216–223, 1994.

[47] Abdolah Chalechale, Golshah Naghdy, and Alfred Mertins. Sketch-based image matching using angular partitioning. *IEEE Trans. Systems, Man, and Cybernetics, Part A*, 35(1):28–41, 2005.

[48] Yonggang Qi, Yi-Zhe Song, Honggang Zhang, and Jun Liu. Sketch-based image retrieval via siamese convolutional neural network. In *2016 IEEE International Conference on Image Processing, ICIP 2016, Phoenix, AZ, USA, September 25-28, 2016*, pages 2460–2464, 2016.

[49] Tu Bui, Leonardo Ribeiro, Moacir Ponti, and John P. Collomosse. Generalisation and sharing in triplet convnets for sketch based visual search. *CoRR*, abs/1611.05301, 2016.

[50] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking. *CoRR*, abs/1404.4661, 2014.

[51] Patsorn Sangkloy, Nathan Burnell, Cusuh Ham, and James Hays. The sketchy database: learning to retrieve badly drawn bunnies. *ACM Trans. Graph.*, 35(4):119:1–119:12, 2016.

[52] Eugenio Di Sciascio, G. Mingolla, and Marina Mongiello. Content-based image retrieval over the web using query by sketch and relevance feedback. In *VISUAL*, volume 1614 of *Lecture Notes in Computer Science*, pages 123–130. Springer, 1999.

[53] Yong Jae Lee, C. Lawrence Zitnick, and Michael F. Cohen. Shadowdraw: real-time user guidance for freehand drawing. *ACM Trans. Graph.*, 30(4):27:1–27:10, 2011.

[54] John P. Collomosse, Graham McNeill, and Leon Adam Watts. Free-hand sketch grouping for video retrieval. In *ICPR*, pages 1–4. IEEE Computer Society, 2008.

[55] David Gavilan Ruiz, Suguru Saito, and Masayuki Nakajima. Sketch-to-collage. In *SIGGRAPH Posters*, page 35. ACM, 2007.

[56] Ricardo Jota, Alfredo Ferreira, Mariana Cerejo, José Santos, Manuel J. Fonseca, and Joaquim A. Jorge. Recognizing hand gestures with CALI. In *SIACG*, pages 187–193. Eurographics Association, 2006.

[57] M. J. Fonseca, S. James, and J. Collomosse. Skeletons from sketches of dancing poses. In *2012 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 247–248, Sept 2012.

[58] Kimmel R. Bronstein M. A., Bronstein M. M. Topology-invariant similarity of nonrigid shapes. *Int. J. Comput. Vis.*, 81:281 – 301, 2009.

[59] Robert Osada, Thomas Funkhouser, Bernard Chazelle, and David Dobkin. Shape distributions. *ACM Transactions on Graphics*, 21:807–832, 2002.

[60] Raif M. Rustamov. Robust volumetric shape descriptor. In *3DOR*, pages 1–5. Eurographics Association, 2010.

[61] Yann Lecun, Lon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998.

[62] Vandeborre J.P. Ansary T.F., Daoudi M. A bayesian 3-d search engine using adaptive views clustering. *IEEE Transactions on Multimedia*, 9(1):78 – 88, 2007.

[63] Biao Leng, Yu Liu, Kai Yu, Xiangyang Zhang, and Zhang Xiong. 3d object understanding with 3d convolutional neural networks. *Inf. Sci.*, 366(C):188–201, October 2016.

[64] Bo Li, Yijuan Lu, Henry Johan, and Ribel Fares. Sketch-based 3d model retrieval utilizing adaptive view clustering and semantic information. *Multimedia Tools Appl.*, 76(24):26603–26631, December 2017.

[65] Thomas Funkhouser, Patrick Min, Michael Kazhdan, Joyce Chen, Alex Halderman, David Dobkin, and David Jacobs. A search engine for 3d models. *ACM Trans. Graph.*, 22(1):83–105, January 2003.

[66] HyoJong Shin and Takeo Igarashi. Magic canvas: Interactive design of a 3-d scene prototype from freehand sketches. In *Proceedings of Graphics Interface 2007*, GI '07, pages 63–70, New York, NY, USA, 2007. ACM.

[67] Kun Xu, Kang Chen, Hongbo Fu, Wei-Lun Sun, and Shi-Min Hu. Sketch2scene: Sketch-based co-retrieval and co-placement of 3d models. *ACM Transactions on Graphics*, 32(4):123:1–123:12, 2013.

[68] Bo Li, Yijuan Lu, Fuqing Duan, Shuilong Dong, Yachun Fan, Lu Qian, Hamid Laga, Haisheng Li, Yuxiang Li, Peng Liu, Maks Ovsjanikov, Hedi Tabia, Yuxiang Ye, Huanpu Yin, and Ziyu Xue. 3d sketch-based 3d shape retrieval. In *Proceedings of the Eurographics 2016 Workshop on 3D Object Retrieval*, 3DOR '16, pages 47–54, Goslar Germany, Germany, 2016. Eurographics Association.

[69] Yuxiang Ye, Bo Li, and Yijuan Lu. 3d sketch-based 3d model retrieval with convolutional neural network. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 2936–2941. IEEE, 2016.

[70] Fang Wang, Le Kang, and Yi Li. Sketch-based 3d shape retrieval using convolutional neural networks. *CoRR*, abs/1504.03504, 2015.

[71] Gen Nishida, Ignacio Garcia-Dorado, Daniel G. Aliaga, Bedrich Benes, and Adrien Bousseau. Interactive sketching of urban procedural models. *ACM Trans. Graph.*, 35(4):130:1–130:11, July 2016.

[72] James H. Clark. Designing surfaces in 3-d. *Commun. ACM*, 19(8):454–460, August 1976.

[73] Jeff Butterworth, Andrew Davidson, Stephen Hench, and Marc. T. Olano. 3dm: A three dimensional modeler using a head-mounted display. In *Proceedings of the 1992 Symposium on Interactive 3D Graphics*, I3D '92, pages 135–138, New York, NY, USA, 1992. ACM.

[74] Arthur W. Brody and Chris Hartman. BLUI: a body language user interface for 3d gestural drawing. In *Human Vision and Electronic Imaging*, volume 3644 of *SPIE Proceedings*, pages 356–363. SPIE, 1999.

[75] Daniel F. Keefe, Robert C. Zeleznik, and David H. Laidlaw. Drawing on air: Input techniques for controlled 3D line illustration, 2007.

[76] Gerold Wesche and Hans-Peter Seidel. Freedrawer: A free-form sketching system on the responsive workbench. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, VRST '01, pages 167–174, New York, NY, USA, 2001. ACM.

[77] Michael F. Deering. The holosketch vr sketching system. *Commun. ACM*, 39(5):54–61, May 1996.

[78] Steven Schkolne, Michael Pruett, and Peter Schröder. Surface drawing: Creating organic 3d shapes with the hand and tangible tools. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '01, pages 261–268, New York, NY, USA, 2001. ACM.

[79] Martins N. V. Beatriz S. Air-Sketching for Object Retrieval. Technical report, Instituto Superior Tecnico, Lisboa, Portugao, 05 2015.

[80] Bo Li, Yijuan Lu, Fuqing Duan, Shuilong Dong, Yachun Fan, Lu Qian, Hamid Laga, Haisheng Li, Yuxiang Li, Peng Liu, Maks Ovsjanikov, Hedi Tabia, Yuxiang Ye, Huanpu Yin, and Ziyu Xue. 3d sketch-based 3d shape retrieval. In A. Ferreira, A. Giachetti, and D. Giorgi, editors, *Eurographics Workshop on 3D Object Retrieval*. The Eurographics Association, 2016.

[81] Gerold Wesche and Hans-Peter Seidel. Freedrawer: A free-form sketching system on the responsive workbench. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, VRST '01, pages 167–174, New York, NY, USA, 2001. ACM.

[82] Michael F. Deering. Holosketch: A virtual reality sketching/animation tool. *ACM Trans. Comput.-Hum. Interact.*, 2(3):220–238, September 1995.

[83] Ryan Schmidt, Azam Khan, Gord Kurtenbach, and Karan Singh. On expert performance in 3d curve-drawing tasks. In *Proceedings of the 6th Eurographics Symposium on Sketch-Based Interfaces and Modeling*, SBIM '09, pages 133–140, New York, NY, USA, 2009. ACM.

[84] E. Wiese, J. H. Israel, A. Meyer, and S. Bongartz. Investigating the learnability of immersive free-hand sketching. In *Proceedings of the Seventh Sketch-Based Interfaces and Modeling Symposium*, SBIM '10, pages 135–142, Aire-la-Ville, Switzerland, Switzerland, 2010. Eurographics Association.

[85] Rahul Arora, Rubaiat Habib Kazi, Fraser Anderson, Tovi Grossman, Karan Singh, and George Fitzmaurice. Experimental evaluation of sketching on surfaces in vr. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 5643–5654, New York, NY, USA, 2017. ACM.

[86] Philipp Wacker, Oliver Nowak, Simon Voelker, and Jan Borchers. Arpen: Mid-air object manipulation techniques for a bimanual ar system with pen and smartphone. In *Proceedings of the*

2019 CHI Conference on Human Factors in Computing Systems, CHI '19, pages 619:1–619:10, New York, NY, USA, May 2019. ACM.

[87] Nianteng Feng, Prakhar Jaiswal, and Rahul Rai. Sketch beautification in air. In *ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers Digital Collection, 2015.

[88] S. Sree Shankar and Rahul Rai. Sketching in three dimensions: A beautification scheme. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 31(3):376–392, 2017.

[89] Doug A. Bowman, Jean Wineman, Larry F. Hodges, and Don Allison. Designing animal habitats within an immersive VE. *IEEE Computer Graphics and Applications*, 18(5):9–13, 1998.

[90] Doug A. Bowman and Chadwick A. Wingrave. Design and evaluation of menu systems for immersive virtual environments. In *Virtual Reality 2001 Conference, VR'01, Yokohama, Japan, March 13-17, 2001, Proceedings*, pages 149–156, 2001.

[91] Zsolt Szalavári and Michael Gervautz. The personal interaction panel - a two-handed interface for augmented reality. *Computer Graphics Forum*, 16(3):335–346, 1997.

[92] Rahul Arora, Rubaiat Habib Kazi, Tovi Grossman, George Fitzmaurice, and Karan Singh. Symbiosissketch: Combining 2d & 3d sketching for designing detailed 3d objects in situ. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 185:1–185:15, New York, NY, USA, 2018. ACM.

[93] Mayra D. Barrera Machuca, Paul Asente, Wolfgang Stuerzlinger, Jingwan Lu, and Byungmoon Kim. Multiplanes: Assisted freehand vr sketching. In *Proceedings of the Symposium on Spatial User Interaction*, SUI '18, pages 36–47, New York, NY, USA, 2018. ACM.

[94] Philipp Wacker, Adrian Wagner, Simon Voelker, and Jan Borchers. Physical guides: An analysis of 3d sketching performance on physical objects in augmented reality. In *Proceedings of the Symposium on Spatial User Interaction*, SUI '18, pages 25–35, New York, NY, USA, 2018. ACM.

[95] Toms Dorta, Gokce Kinayoglu, and Michael Hoffmann. Hyve-3d and the 3d cursor: Architectural co-design with freedom in virtual reality. *International Journal of Architectural Computing*, 14(2):87–102, 2016.

[96] Mandayam A. Srinivasan and Cagatay Basdogan. Haptics in virtual environments: Taxonomy, research status, and challenges. *Computers & Graphics*, 21(4):393–404, 1997.

[97] Thomas H. Massie and J. Kenneth Salisbury. The PHANToM haptic interface: A device for probing virtual objects. In *Proceedings of the ASME Dynamic Systems and Control Division*, pages 295–301, 1994.

[98] Chris Raymaekers, Gert Vansichem, and Frank Van Reeth. Improving sketching by utilizing haptic feedback. In *AAAI spring symposium on sketch understanding*, pages 113–117, 2002.

[99] Brent Edward Insko. *Passive Haptics Significantly Enhances Virtual Environments*. PhD thesis, University of North Carolina at Chapel Hill, USA, 2001.

[100] Anton Franzluebbers and Kyle Johnsen. Performance benefits of high-fidelity passive haptic feedback in virtual reality training. In *Proceedings of the Symposium on Spatial User Interaction*, SUI '18, pages 16–24, New York, NY, USA, 2018. ACM.

[101] André Zenner and Antonio Krüger. Shifty: A weight-shifting dynamic passive haptic proxy to enhance object perception in virtual reality. *IEEE Transactions on Visualization and Computer Graphics*, 23(4):1285–1294, 2017.

[102] Eric Whitmire, Hrvoje Benko, Christian Holz, Eyal Ofek, and Mike Sinclair. Haptic revolver: Touch, shear, texture, and shape rendering on a reconfigurable virtual reality controller. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 86:1–86:12, New York, NY, USA, 2018. ACM.

[103] Bruno Araujo, Ricardo Jota, Varun Perumal, Jia Xian Yao, Karan Singh, and Daniel Wigdor. Snake charmer: Physically enabling virtual objects. In *Proceedings of the TEI '16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction*, TEI '16, pages 218–226, New York, NY, USA, 2016. ACM.

[104] Philip R. Cohen, Michael Johnston, David McGee, Sharon Oviatt, Jay Pittman, Ira Smith, Liang Chen, and Josh Clow. Quickset: Multimodal interaction for distributed applications. In *Proceedings of the Fifth ACM International Conference on Multimedia*, MULTIMEDIA '97, pages 31–40, New York, NY, USA, 1997. ACM.

[105] Jiagen Jin and Wenfeng Li. A survey of the information fusion in mmhci. In *2010 International Conference on Machine Vision and Human-machine Interface*, pages 509–513. IEEE, 2010.

[106] Alex Pentland. Smart rooms, smart clothes. In *Proceedings. Fourteenth International Conference on Pattern Recognition (Cat. No. 98EX170)*, volume 2, pages 949–953. IEEE, 1998.

[107] Minh Tue Vo and Cindy Wood. Building an application framework for speech and pen input integration in multimodal learning interfaces. In *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, volume 6, pages 3545–3548. IEEE, 1996.

[108] Nikolaos G Tsagarakis, John O Gray, Darwin G Caldwell, Cinzia Zannoni, Marco Petrone, Debora Testi, and Marco Viceconti. A haptic-enabled multimodal interface for the planning of hip arthroplasty. *IEEE MultiMedia*, 13(3):40–48, 2006.

[109] LWJ Boves and EA Den Os. Must-multimodal and multilingual services for small mobile terminals, 2002.

[110] Wolfgang Wahlster. *SmartKom: foundations of multimodal dialogue systems*, volume 12. Springer, 2006.

[111] Boris W van Schooten, R Op Den Akker, Sophie Rosset, Olivier Galibert, Aurelien Max, and Gabriel Illouz. Follow-up question handling in the imix and ritel systems: A comparative study. *Natural Language Engineering*, 15(1):97–118, 2009.

[112] P. R. Cohen, M. Dalrymple, D. B. Moran, F. C. Pereira, and J. W. Sullivan. Synergistic use of direct manipulation and natural language. *SIGCHI Bull.*, 20(SI):227–233, March 1989.

[113] Peter Kay. Speech-driven graphics: a user interface. *Journal of Microcomputer Applications*, 16(3):223–231, 1993.

[114] Anil Jain, Karthik Nandakumar, and Arun Ross. Score normalization in multimodal biometric systems. *Pattern recognition*, 38(12):2270–2285, 2005.

[115] Aaron Adler and Randall Davis. Speech and sketching for multimodal design. In *ACM SIGGRAPH 2007 Courses*, SIGGRAPH '07, New York, NY, USA, 2007. ACM.

[116] A. Adler and R. Davis. Speech and sketching: An empirical study of multimodal interaction. In *Proceedings of the 4th Eurographics Workshop on Sketch-based Interfaces and Modeling*, SBIM '07, pages 83–90, New York, NY, USA, 2007. ACM.

[117] Jason Linder, Gierad Laput, Mira Dontcheva, Gregg Wilensky, Walter Chang, Aseem Agarwala, and Eytan Adar. Pixeltone: a multimodal interface for image editing. In *2013 ACM SIGCHI*

Conference on Human Factors in Computing Systems, CHI '13, Paris, France, April 27 - May 2, 2013, Extended Abstracts, pages 2829–2830, 2013.

[118] Jürgen M. Janas. The semantics-based natural language interface to relational databases. In Leonard Bolc and Matthias Jarke, editors, *Cooperative Interfaces to Information Systems*, pages 143–188, Berlin, Heidelberg, 1986. Springer Berlin Heidelberg.

[119] Fei Li and Hosagrahar V Jagadish. Nalir: An interactive natural language interface for querying relational databases. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, SIGMOD '14, pages 709–712, New York, NY, USA, 2014. ACM.

[120] David L. Waltz. An english language question answering system for a large relational database. *Commun. ACM*, 21(7):526–539, July 1978.

[121] Alexander Gruenstein, Bo-June Paul Hsu, James Glass, Stephanie Seneff, Lee Hetherington, Scott Cyphers, Ibrahim Badr, Chao Wang, and Sean Liu. A multimodal home entertainment interface via a mobile device. In *Proceedings of the ACL-08: HLT Workshop on Mobile Language Processing*, pages 1–9, Columbus, Ohio, June 2008. Association for Computational Linguistics.

[122] Lina Zhou, Mohammedammar Shaikh, and Dongsong Zhang. Natural language interface to mobile devices. In Zhongzhi Shi and Qing He, editors, *Intelligent Information Processing II*, pages 283–286, Boston, MA, 2005. Springer US.

[123] Shinya Kikuchi and Partha Chakroborty. Car-following model based on fuzzy inference system. *Transportation Research Record*, pages 82–82, 1992.

[124] Niels Ole Bernsen and Laila Dybkjr. Exploring natural interaction in the car. In *Proceedings of the International Workshop on Information Presentation and Natural Multimodal Dialogue*, pages 75–79, 2001.

[125] Gustavo López, Luis Quesada, and Luis A. Guerrero. Alexa vs. siri vs. cortana vs. google assistant: A comparison of speech-based natural user interfaces. In Isabel L. Nunes, editor, *Advances in Human Factors and Systems Interaction*, pages 241–250, Cham, 2018. Springer International Publishing.

[126] Amrita S. Tulshan and Sudhir Namdeorao Dhage. Survey on virtual assistant: Google assistant, siri, cortana, alexa. In Sabu M. Thampi, Oge Marques, Sri Krishnan, Kuan-Ching Li, Domenico

Ciuonzo, and Maheshkumar H. Kolekar, editors, *Advances in Signal Processing and Intelligent Recognition Systems*, pages 190–201, Singapore, 2019. Springer Singapore.

[127] Sean Li and Xiaojun (Jenny) Yuan. A review of the current intelligent personal agents. In Constantine Stephanidis, editor, *HCI International 2018 – Posters' Extended Abstracts*, pages 253–257, Cham, 2018. Springer International Publishing.

[128] Scott McGlashan. Speech interfaces to virtual reality. In *Proceedings of 2nd International Workshop on Military Applications of Synthetic Environments and Virtual Reality*, 1995.

[129] Scott McGlashan and Tomas Axling. A speech interface to virtual environments. In *Proc., International Workshop on Speech and Computers*, 1996.

[130] John H. L. Hansen, Rongqing Huang, Bowen Zhou, Michael Seadle, John Robert Deller, Aparna Gurijala, Mikko Kurimo, and Pongtep Angkititrakul. Speechfind: spoken document retrieval for a national gallery of the spoken word. *Proceedings of the 6th Nordic Signal Processing Symposium, 2004. NORSIG 2004.*, pages 1–4, 2004.

[131] Masataka Goto, Jun Ogata, and Kouichirou Eto. Podcastle: a web 2.0 approach to speech recognition research. In *INTERSPEECH 2007, 8th Annual Conference of the International Speech Communication Association, Antwerp, Belgium, August 27-31, 2007*, pages 2397–2400, 2007.

[132] C. Alberti, M. Bacchiani, A. Bezman, C. Chelba, A. Drofa, H. Liao, P. Moreno, T. Power, A. Sahuguet, M. Shugrina, and O. Siohan. An audio indexing system for election video material. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4873–4876, April 2009.

[133] James R. Glass, Timothy J. Hazen, D. Scott Cyphers, Igor Malioutov, David Huynh, and Regina Barzilay. Recent progress in the MIT spoken lecture processing project. In *INTERSPEECH 2007, 8th Annual Conference of the International Speech Communication Association, Antwerp, Belgium, August 27-31, 2007*, pages 2553–2556, 2007.

[134] Lin shan Lee Sheng-yi Kong, Miao ru Wu, Che kuang Lin, Yi sheng Fu, Yungyu Chung, Yu Huang, and Yun-Nung Chen. Ntu virtual instructor - a spoken language system o ering services of learning on demand using video/audio/slides of course lectures. *ICASSP*, 2009.

[135] Ye-Yi Wang, Dong Yu, Yun-Cheng Ju, and Alex Acero. An introduction to voice search. *IEEE Signal Processing Magazine*, 25(3):28–38, 2008.

[136] Alexander G Hauptmann. Speech and gestures for graphic image manipulation. In *ACM SIGCHI Bulletin*, volume 20, pages 241–245. ACM, 1989.

[137] J LaViola. Whole-hand and speech input in virtual environments. *Unpublished masters thesis, Department of Computer Science, Brown University, CS-99-15*, 1999.

[138] Jan Ciger, Mario Gutierrez, Frederic Vexo, and Daniel Thalmann. The magic wand. In *Proceedings of the 19th Spring Conference on Computer Graphics*, SCCG '03, pages 119–124, New York, NY, USA, 2003. ACM.

[139] H.R. Sheikh, M.F. Sabir, and A.C. Bovik. A statistical evaluation of recent full reference image quality assessment algorithms. *IEEE Trans. on Image Processing*, 15(11):3440–3451, 2006.

[140] Nikolay Ponomarenko, Lina Jin, Oleg Ieremeiev, Vladimir Lukin, Karen Egiazarian, Jaakko Astola, Benoit Vozel, Kacem Chehdi, Marco Carli, Federica Battisti, and C.-C. Jay Kuo. Image database TID2013: Peculiarities, results and perspectives. *Signal Processing: Image Communication*, 30:57–77, Jan. 2015.

[141] Scott J Daly. Visible differences predictor: An algorithm for the assessment of image fidelity. In *Human Vision, Visual Processing, and Digital Display III*, volume 1666, pages 2–16. International Society for Optics and Photonics, 1992.

[142] Tunç Ozan Aydin, Rafał Mantiuk, Karol Myszkowski, and Hans-Peter Seidel. Dynamic range independent image quality assessment. *ACM Transactions on Graphics (TOG)*, 27(3):69, 2008.

[143] Nikolay Ponomarenko, Oleg Eremeev, Vladimir Lukin, and Karen Egiazarian. Statistical evaluation of no-reference image visual quality metrics. In *2010 2nd European Workshop on Visual Information Processing (EUVIP)*, pages 50–54, 2010.

[144] L. Zhang, Y. Shen, and H. Li. Vsi: A visual saliency-induced index for perceptual image quality assessment. *IEEE Transactions on Image Processing*, 23(10):4270–4281, 2014.

[145] L. Zhang, L. Zhang, X. Mou, and D. Zhang. Fsim: A feature similarity index for image quality assessment. *IEEE Transactions on Image Processing*, 20(8):2378–2386, 2011.

[146] Weisi Lin and C.-C. Jay Kuo. Perceptual visual quality metrics: A survey. *J. Visual Communication and Image Representation*, pages 297–312, 2011.

[147] Damon M A Chandler. Seven challenges in image quality assessment: Past, present, and future research. *ISRN Signal Processing*, 2013(17), 2013.

[148] X. Zhang and B. A. Wandell. A spatial extension of CIELAB for digital color-image reproduction. *Journal of the Society for Information Display*, 5(1):61, 1997.

[149] Scott Daly. The Visible Differences Predictor: An algorithm for the assessment of image fidelity. In *Digital Images and Human Vision*, pages 179–206. MIT Press, 1993.

[150] Jeffrey Lubin. *Vision models for target detection and recognition*, chapter A Visual Discrimination Model for Imaging System Design and Evaluation, pages 245–283. World Scientific, 1995.

[151] Martin Čadík, Robert Herzog, Rafał K. Mantiuk, Karol Myszkowski, and Hans-Peter Seidel. New measurements reveal weaknesses of image quality metrics in evaluating graphics artifacts. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, 31(6):147, 2012.

[152] Jyrki Alakuijala, Robert Obryk, Ostap Stoliarchuk, Zoltan Szabadka, Lode Vandevenne, and Jan Wassenberg. Guetzli: Perceptually guided JPEG encoder. *arXiv:1703.04421*, 2017.

[153] M. M. Alam, K. P. Vilankar, David J Field, and Damon M Chandler. Local masking in natural images: A database and analysis. *Journal of Vision*, 14(8):22, jul 2014.

[154] Robert Herzog, Martin Čadík, Tunç O. Aydin, Kwang In Kim, Karol Myszkowski, and Hans-Peter Seidel. Norm: No-reference image quality metric for realistic image synthesis. *Computer Graphics Forum*, 31(2):545–554, 2012.

[155] Martin Čadík, Robert Herzog, Rafał Mantiuk, Radosław Mantiuk, Karol Myszkowski, and Hans-Peter Seidel. Learning to predict localized distortions in rendered images. In *Computer Graphics Forum*, volume 32, pages 401–410, 2013.

[156] Simone Bianco, Luigi Celona, Paolo Napoletano, and Raimondo Schettini. On the use of deep learning for blind image quality assessment. *arXiv:1602.05531*, 2016.

[157] Le Kang, Peng Ye, Yi Li, and David Doermann. Convolutional neural networks for no-reference image quality assessment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1733–1740, 2014.

[158] Sebastian Bosse, Dominique Maniry, Thomas Wiegand, and Wojciech Samek. A deep neural network for image quality assessment. In *IEEE International Conference on Image Processing (ICIP)*, pages 3773–3777. IEEE, 2016.

[159] Sebastian Bosse, Dominique Maniry, Thomas Wiegand, and Wojciech Samek. Neural network-based full-reference image quality assessment. In *Proceedings of the Picture Coding Symposium (PCS)*, pages 1–5, 2016.

[160] Manish Narwaria and Weisi Lin. Objective image quality assessment based on support vector regression. *IEEE Transactions on Neural Networks*, 21(3):515–9, 2010.

[161] A.K. Moorthy and A.C. Bovik. A two-step framework for constructing blind image quality indices. *IEEE Signal Processing Letters*, 17(5):513 –516, 2010.

[162] M.A. Saad, A.C. Bovik, and C. Charrier. Blind image quality assessment: A natural scene statistics approach in the DCT domain. *IEEE Trans. on Image Processing*, 21(8):3339–3352, 2012.

[163] Huixuan Tang, N. Joshi, and A. Kapoor. Learning a blind measure of perceptual image quality. *Proc. of IEEE Computer Vision and Pattern Recognition*, pages 305–312, 2011.

[164] S. Bosse, D. Maniry, K. R. Müller, T. Wiegand, and W. Samek. Deep neural networks for no-reference and full-reference image quality assessment. *IEEE Trans. Image Proc.*, 27(1):206–219, 2018.

[165] Seyed Ali Amirshahi, Marius Pedersen, and Stella X Yu. Image quality assessment by comparing cnn features between images. *Journal of Imaging Science and Technology*, 60(6):60410–1, 2016.

[166] Daniele Giunchi, Stuart James, Donald Degraen, and Anthony Steed. Mixing realities for sketch retrieval in virtual reality. In *The 17th International Conference on Virtual-Reality Continuum and Its Applications in Industry*, VRCAI 19, New York, NY, USA, 2019. Association for Computing Machinery.

[167] D. Giunchi, S. James, and A. Steed. Model retrieval by 3d sketching in immersive virtual reality. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 559–560, 2018.

[168] Daniele Giunchi, Stuart James, and Anthony Steed. 3d sketching for interactive model retrieval in virtual reality. In *Proceedings of the Joint Symposium on Computational Aesthetics and Sketch-Based Interfaces and Modeling and Non-Photorealistic Animation and Rendering*, Expressive '18, pages 1:1–1:12, New York, NY, USA, 2018. ACM.

[169] Avery Wang. The shazam music recognition service. *Commun. ACM*, 49(8):4448, August 2006.

[170] Mathias Eitz, Kristian Hildebrand, Tamy Boubekeur, and Marc Alexa. Sketch-based image retrieval: Benchmark and bag-of-features descriptors. *IEEE Transactions on Visualization and Computer Graphics*, 17(11):1624–1636, 2011.

[171] Manuel J. Fonseca, Alfredo Ferreira, and Joaquim A. Jorge. Towards 3d modeling using sketches and retrieval. In *Proceedings of the First Eurographics Conference on Sketch-Based Interfaces and Modeling*, SBM'04, pages 127–136, Aire-la-Ville, Switzerland, Switzerland, 2004. Eurographics Association.

[172] T. Bui, L. Ribeiro, M. Ponti, and J. Collomosse. Compact descriptors for sketch-based image retrieval using a triplet loss convolutional neural network. *Computer Vision and Image Understanding*, 2017.

[173] Q. Yu, F. Liu, Y. Z. Song, T. Xiang, T. M. Hospedales, and C. C. Loy. Sketch me that shoe. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 799–807, June 2016.

[174] Tiago Santos, Alfredo Ferreira, Filipe Dias, and Manuel J. Fonseca. Using sketches and retrieval to create lego models. In *Proceedings of the Fifth Eurographics Conference on Sketch-Based Interfaces and Modeling*, SBM'08, pages 89–96, Aire-la-Ville, Switzerland, Switzerland, 2008. Eurographics Association.

[175] Manuel J. Fonseca, Alfredo Ferreira, and Joaquim A. Jorge. Towards 3d modeling using sketches and retrieval. In *Proceedings of the First Eurographics Conference on Sketch-Based Interfaces and Modeling*, SBM'04, pages 127–136, Aire-la-Ville, Switzerland, Switzerland, 2004. Eurographics Association.

[176] Tu Bui, Leonardo Ribeiro, Moacir Ponti, and John Collomosse. Compact descriptors for sketch-based image retrieval using a triplet loss convolutional neural network. *Computer Vision and Image Understanding*, 164:27–37, 2017.

[177] Kaiyue Pang, Ke Li, Yongxin Yang, Honggang Zhang, Timothy M Hospedales, Tao Xiang, and Yi-Zhe Song. Generalising fine-grained sketch-based image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 677–686, 2019.

[178] Yonggang Qi, Yi-Zhe Song, Honggang Zhang, and Jun Liu. Sketch-based image retrieval via siamese convolutional neural network. In *2016 IEEE International Conference on Image Processing, ICIP 2016, Phoenix, AZ, USA, September 25-28, 2016* [48], pages 2460–2464.

[179] Bo Li, Yijuan Lu, Azeem Ghumman, Bradley Strylowski, Mario Gutierrez, Safiyah Sadiq, Scott Forster, Natacha Feola, and Travis Bugerin. 3d sketch-based 3d model retrieval. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, ICMR '15, page 555558, New York, NY, USA, 2015. Association for Computing Machinery.

[180] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik G. Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proc. International Conference on Computer Vision (ICCV)*, 2015.

[181] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *British Machine Vision Conference*, 2014.

[182] Michele Fiorentino, Giuseppe Monno, Pietro A Renzulli, and Antonio E Uva. 3d sketch stroke segmentation and fitting in virtual reality. In *International conference on the Computer Graphics and Vision*, volume 5, 2003.

[183] Dominik Rausch, Ingo Assenmacher, and Torsten W. Kuhlen. 3d sketch recognition for interaction in virtual environments. In *Proceedings of the Seventh Workshop on Virtual Reality Interactions and Physical Simulations, VRIPHYS 2010, Copenhagen, Denmark, 2010.*, pages 115–124, 2010.

[184] Han-wool Choi, Hee-joon Kim, Jeong-in Lee, and Young-Ho Chai. Free hand stroke based virtual sketching, deformation and sculpting of nurbs surface. In *Proceedings of the 2005 International Conference on Augmented Tele-existence*, ICAT '05, pages 3–9, New York, NY, USA, 2005. ACM.

[185] Stuart James and John Collomosse. Interactive video asset retrieval using sketched queries. In *Proceedings of the 11th European Conference on Visual Media Production*, CVMP '14, pages 11:1–11:8, New York, NY, USA, 2014. ACM.

[186] Alaa E. Abdel-Hakim and Aly A. Farag. CSIFT: A SIFT descriptor with color invariant characteristics. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006), 17-22 June 2006, New York, NY, USA*, pages 1978–1983, 2006.

[187] Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. *CoRR*, abs/1712.04621, 2017.

[188] Connor Shorten and Taghi M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60, Jul 2019.

[189] Sharon Oviatt. User-centered modeling and evaluation of multimodal interfaces. *Proceedings of the IEEE*, 91(9):1457–1468, 9 2003.

[190] Richard Socher, Cliff C Lin, Chris Manning, and Andrew Y Ng. Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 28th ICML*, pages 129–136, 2011.

[191] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th ICML*, volume 28 of *Proceedings of Machine Learning Research*, pages 1310–1318, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

[192] Klaus Greff, Rupesh Kumar Srivastava, Jan Koutník, Bas R. Steunebrink, and Jürgen Schmidhuber. LSTM: A search space odyssey. *IEEE Trans. Neural Networks Learn. Syst.*, 28(10):2222–2232, 2017.

[193] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008, 2017.

[194] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. pages 1097–1105, 2012.

[195] Krzysztof Wolski, Daniele Giunchi, Nanyang Ye, Piotr Didyk, Karol Myszkowski, Radosław Mantiuk, Hans-Peter Seidel, Steed Anthony, and Rafał K. Mantiuk. Dataset and metrics for predicting local visible differences. *ACM Transactions on Graphics (ToG)*, 2018.

[196] Rafał Piórkowski, Radosław Mantiuk, and Adam Siekawa. Automatic detection of game engine artifacts using full reference image quality metrics. *ACM Transactions on Applied Perception (TAP)*, 14(3):14, 2017.

[197] Kanita Karađuzović-Hadžiabdić, Jasminka Hasić Telalović, and Rafał K Mantiuk. Assessment of multi-exposure hdr image deghosting methods. *Computers & Graphics*, 63:1–17, 2017.

[198] Vamsi K. Adhikarla, Marek Vinkler, Denis Sumin, Rafal K. Mantiuk, Karol Myszkowski, Hans-Peter Seidel, and Piotr Didyk. Towards a quality metric for dense light fields. In *Computer Vision and Pattern Recognition*, 2017.

[199] Sebastian Bosse, Dominique Maniry, Klaus-Robert Mueller, Thomas Wiegand, and Wojciech Samek. Full-reference image quality assessment using neural networks. In *Int. Work. Qual. Multimdia Exp.*, 2016.

[200] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[201] S. S. Panda, M. S. R. S. Prasad, and G. Jena. POCS based super-resolution image reconstruction using an adaptive regularization parameter. *CoRR*, abs/1112.1484, 2011.

[202] 3Doodler. `https://intl.the3doodler.com/`.

[203] Daniel Dixon, Manoj Prasad, and Tracy Hammond. icandraw: Using sketch recognition and corrective feedback to assist a user in drawing human faces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 897–906, New York, NY, USA, 2010. ACM.

[204] Jungwoo Choi, Heeryon Cho, Jinjoo Song, and Sang Min Yoon. Sketchhelper: Real-time stroke guidance for freehand sketch retrieval. *IEEE Transactions on Multimedia*, 21(8):2083–2092, 2019.

[205] Krzysztof Wolski, Daniele Giunchi, Shin-ichi Kinuwaki, Piotr Didyk, Karol Myszkowski, Anthony Steed, and Rafal Mantiuk. Selecting texture resolution using a task-specific visibility metric. *Comput. Graph. Forum*, 38(7):685–696, 2019.