Universidad Zaragoza

Master's thesis

Deep learning models for 3D mesh saliency prediction

Author

Andrés Fandos Villanueva

Supervisors

Belén Masiá Corcoy

Ana Serrano Pacheu

Master in Robotics, Graphics and Computer Vision

2021/2022

Escuela de Ingeniería y Arquitectura Universidad Zaragoza

TRABAJOS DE FIN DE GRADO / FIN DE MÁSTER

Escuela de Ingeniería y Arquitectura Universidad Zaragoza

DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe remitirse a seceina@unizar.es dentro del plazo de depósito)

D./Dª. Andrés Fandos Villanueva ,

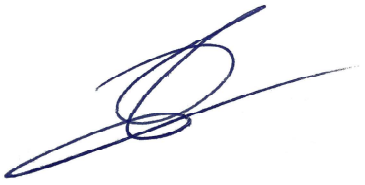
en aplicación de lo dispuesto en el art. 14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de Estudios de la titulación de Máster Universitario en Robótica, Gráficos y Visión por Computador (Título del Trabajo)

Deep learning models for 3D mesh saliency prediction

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, 21/06/2022



Fdo: Andrés Fandos Villanueva

Abstract

Immersive applications and virtual environments are new technologies with more influence every day in society. This technologies consist of computer-generated environments with scenes and objects that appear to be real, making the users feel they are immersed in their surroundings. Instead of seeing a screen in front of them, users immerse themselves and interact with 3D worlds. In these environments, since the user controls everything (position, gaze, interaction), each person can interpret and perceive the same content differently, making it much more difficult for content creators to direct the user's attention where they want them to look. It is for this reason that predicting how humans behave in virtual environments remains an open problem. This understanding and modelling of the human visual attention behaviour is of great importance in some computer vision and computer graphics applications such as foveated rendering compression or content design.

The study of human visual attention has been widely explored in many works. It consists of detecting and identifying the regions of the stimulus that most attract the user's attention, generally using an eye tracker to collect the data. This measure of the regions of interest is known as *saliency*, and its detection and modelling is a fundamental problem in computer graphics and computer vision. In 3D, human visual attention can vary a lot depending on the user and the scene. The same scene can be explored by different people in different ways, resulting in inter-observer variability, and even when a subject sees the same scene twice, different parts of it are likely to be explored, resulting in intra-observer variability. For these reasons, predicting saliency is a very challenging task.

This Master's thesis presents a saliency prediction model for point clouds based on deep learning. To the best of our knowledge, this is the first deep learning approach that predicts saliency on point clouds using real saliency data as ground truth, instead of using point-based methods or methods which study the rarity or curvature of a mesh splitting it in clusters of points. In this project, an experiment with 32 people was carried out in order to collect the gaze data that served as ground truth. Then a deep neural network based on previous work [1] was used and adapted to our problem in order to predict 3D saliency on objects. Finally results were obtained by training this network with the data collected in the experiment. The study proves that the model is able to recognise the most interesting parts of a mesh.

Acknowledgments

I would like to thank my supervisors Belén Masiá and Ana Serrano for the time they dedicated to help and guide me along this project, it has been a great year working on this interesting research project with you. Of course, I also want to thank Daniel Martín for being my supervisor during the internship last year and for helping me with this project as well.

I would also like to thank all members of the *Graphics and Imaging Lab* for receiving and accepting me from the first moment and for dedicating a moment of their lives to participate in my experiment. You are awesome.

Thanks also to all my friends who came to the lab to do the experiment, your participation was very helpful and meant a lot to me.

Finally, I want to thank my family and my girlfriend for supporting me at all times and for being there in the most difficult moments, especially the last two years.

Contents

List of Figures

List of Tables

1. Introduction

1.1 Project context

In the field of computer graphics, 3D visual attention prediction has been an important issue. With the rise of immersive applications and virtual environments, understanding and modelling human visual attention is of great importance in some applications of this field such as foveated rendering compression, 3D content design, face recognition [4], similarity and alignment [5], simplification and icon generation [6], abstraction [7] and viewpoint selection [8].

During the last years the problem of 3D mesh saliency prediction has been widely explored, starting from approaches that base their predictions on statistical algorithms [8, 9, 10, 11], to the most recent approaches that base their predictions on the features extracted within the hidden layers of neural networks designed for classification problems [3, 2, 12]. All these works aim to represent the understanding of 3D surfaces from the perspective that some regions of a 3D surface are more important than others according to human perception. However, most of the available saliency prediction works rely on handcrafted features and descriptors that do not generalize well. As an example, Fig. 1.1 shows a ground truth saliency map created from gaze data collected in an experiment carried out with people by Lavoué et al. [13], along with some predictions using different statistical saliency prediction methods. It can be seen how these methods focus (incorrectly) primarily on parts of the shape with steep curvatures and high-frequency details.

Figure 1.1: Results of 3D mesh saliency prediction using statistical algorithms. From left to right: ground truth from Lavoué et al. [13], saliency prediction by Lee et al. [8], saliency prediction by Leifman et al. [9], saliency prediction by Song et al. [10], saliency prediction by Tasse et al. [11]. Image extracted from Lavoué et al. [13].

On the other hand, the main problem with 3D mesh saliency prediction is that collecting enough gaze data to train a model is a difficult task, and existing mesh saliency datasets are either small [13, 14] or the data has not been collected by eye-tracking experiments but by asking people to manually select points of a mesh that they find interesting [15]. The works that make use of deep learning to predict saliency on meshes [3, 2, 12] do not use real saliency data. These

works use neural networks designed for classification and part segmentation tasks, and, since these deep learning models abstract feature vectors from the meshes along the hidden layers to label each point, their approaches assume that very different feature vectors correspond to salient parts. Fig. 1.2 shows a comparison between Song et al. [2] and Liu et al. [3] against the ground truth created by Chen et al. [15]. It can be seen that these approaches are able to predict saliency reasonably well, but still, as their main assumption to create the saliency maps is that different feature vectors correspond to salient parts, those salient parts are mostly corners, strong curvatures and high-frequency details.

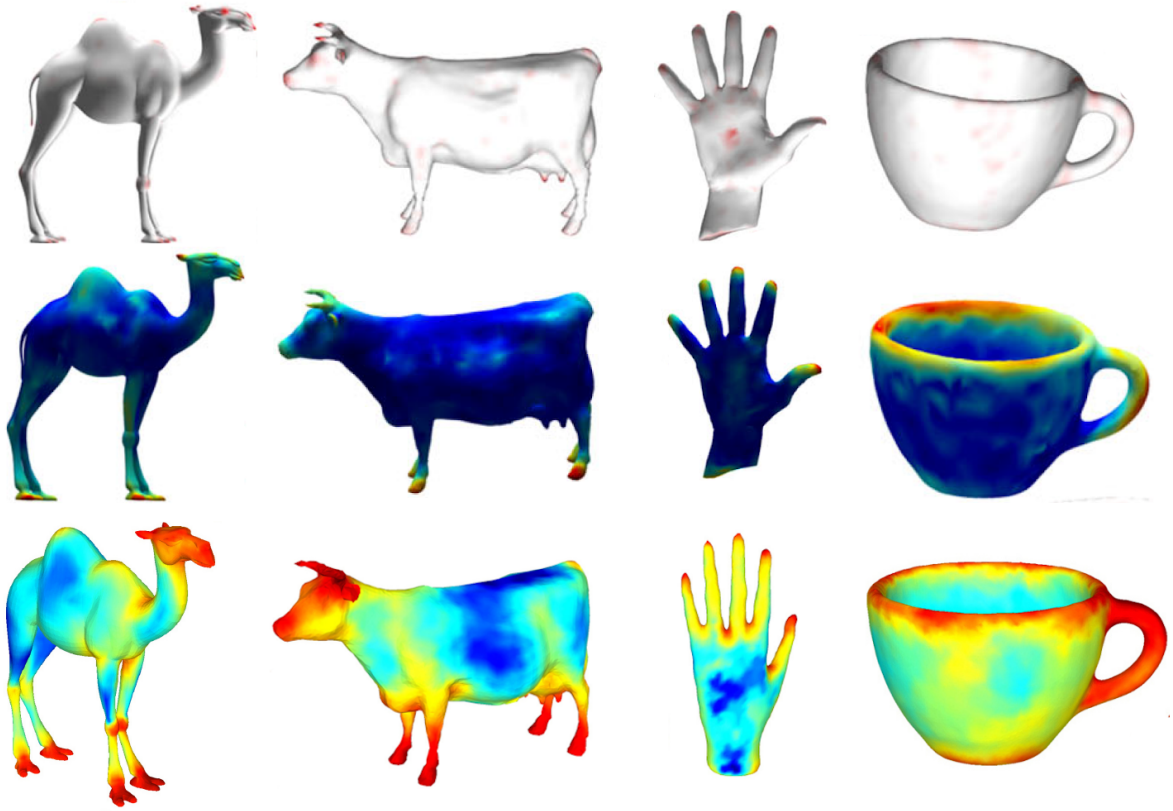


Figure 1.2: Results of 3D mesh saliency prediction using deep learning. Warmer colors represent the salient parts of the objects, i.e. the parts of the objects that are most likely to attract human attention. From top to bottom: ground truth created by Chen et al. [15], saliency prediction results from Liu et al. [3], saliency prediction results from Song et al. [2]. Images in the two first rows were extracted from Liu et al. [3]. Images in the bottom row were extracted from Song et al. [2].

Furthermore, with the rise of neural scene representations [16, 17, 18, 19], saliency prediction can be an interesting research topic along with neural representations in order to generate new points of view of a scene or object and at the same time predict the saliency of that scene or object seen from that point of view.

This project is devoted to deal with the saliency prediction problem on 3D meshes, more specifically on point clouds. At first, a model was designed and trained on existing datasets [13, 14]. However, this data revealed to be insufficient, so more data was gathered through an experiment creating our own dataset, larger than those provided by these two works. To collect the data, an eye tracker was used to record users' gaze while viewing different meshes. After

collecting the data and preparing the dataset, the neural network architecture from Qi et al. [1], with some modifications, was used as a baseline as it has proven to be very efficient and robust when working with point clouds. Finally, the model was evaluated in a qualitative and quantitative fashion.

1.2 Goals and scope of the project

The motivation of this Master's thesis is to face the saliency prediction problem on 3D meshes by making use of deep learning and utilising for the first time real gaze data as ground truth to train the model. The project proposes several goals:

- Studying of the state of the art on modeling human visual attention behavior while visualizing 2D images and 3D objects, on saliency prediction for both 2D and 3D content, and on deep learning algorithms for point clouds (Section 2).
- Conducting a user study for capturing gaze data on 3D meshes. The Unity game engine was used, compatible with virtual reality systems and eye-tracking tools, along with the Pupil Labs[1] eye tracker, so that it was possible to capture and collect gaze data from users while viewing the stimulus (Section 3).
- Adaptation and implementation of a deep learning-based model for saliency prediction on point clouds (Section 4).
- Evaluation of the performance of the model and comparison with other state-of-the-art works that predict saliency on 3D meshes and point clouds (Section 5).

This project has been carried out in the *Graphics and Imaging Lab* research group, at the University of Zaragoza. The group's work focuses on computer graphics, conducting several areas of research such as rendering, image processing, computational imaging, virtual reality, or applied perception, among others. In addition, the group's work frequently involves the use of gaze tracking and deep learning techniques.

1.3 Planning and tools

The timeline followed to achieve the goals described above for this Master's thesis is shown in the Gantt chart in Fig. 1.3. The total time dedicated to the project has been 775 hours distributed throughout the year. The project has been developed iteratively in order to study more efficient approaches and correct the errors that could arise at each stage. Progress was evaluated every week.

The programming language utilised to develop the project has been Python, and, for the implementation of the model, the framework PyTorch has been used. The deep learning model was trained in a NVIDIA RTX 3060 GPU with an integrated memory of 12 GB.

[1] https://pupil-labs.com/

			2021				2022					
Stage	**Task**	**Hours**	**Sept**	**Oct**	**Nov**	**Dec**	**Jan**	**Feb**	**Mar**	**Apr**	**May**	**Jun**
Previous work	Study of the state-of-the-art	120										
Experiment	Prepare scene in Unity	15										
	Search and implement stimuli	30										
	Implement Pupil Labs plugin	20										
	Implement data collection script	40										
	Experiment design	10										
	Validation and verification	15										
	Data protection docs and approval	10										
	Experiments	40										
	Scripts for processing gaze data	45										
	Scripts for metrics from gaze data	20										
Model	Search, train and test models for pointclouds	55										
	Adapt shapes and model chosen to our problem	20										
	Prepare dataset with our data	20										
	Train/test PointNet++ models with our data	150										
Analysis	Evaluation of metrics of the models trained	25										
	Evaluate model and comparison with other works	50										
	Abaltion studies	30										
Documentation	Final master report	60										
	Total hours	**775**										

Figure 1.3: Project timeline Gantt Chart.

2. Related work

In this section, an overview of the state of the art in saliency prediction is provided. Initially, the basics of eye tracking are introduced, since it is the most common way of collecting fixation data to study human visual attention behaviour and to predict saliency. Then, the main works of 2D saliency prediction are explained. Next, the most relevant existing approaches to address the problem of 3D saliency prediction are exposed. And finally, an introduction to some deep learning models that work with point clouds is presented.

2.1 Eye tracking

Modeling and predicting human visual attention behavior has been an area of great interest in vision research. In order to create ground truth saliency datasets in the field of computer vision, it is very common to perform eye-tracking experiments on people. Most of the eye trackers utilized are based on video cameras directed towards the eyes. Usually, the manufacturers of these devices provide the software required to extract the positions of the center of the pupils from each video frame.

Many works have used eye trackers to study and model human visual attention behaviour. Liu et al. [20] explored the guidance and constancy of visualization variables in 3D visualization using eye-tracking techniques. Lee et al. [21] used eye-tracking to understand the effect of specific architectural elements on viewers' visual attention. Howlett et al. [22] captured human gaze data conducting an eye-tracking experiment in order to detect salient features and then investigate if salient features exist and can be predicted in advance. The three works from Wang et al. [23, 24, 14] conducted an eye-tracking experiment: in the first work [23] the experiment was carried out using a monocular head-mounted eye tracker and the goal was to track gaze positions on a three dimensional object, in the second work [24] the experiment was also carried out with a monocular eye tracker but the goal was to corroborate the assumption that the saliency found in flat stimuli can be related to the underlying 3D scene, and finally, in the third work [14] the experiment was carried out using a binocular eye tracker and the goal was to provide the first large dataset of human fixations on physical 3D objects, varying the point of view and made of different materials, and to analyze the similarities among the different conditions. Lavoué et al. [13] conducted two eye-tracking experiments involving 3D shapes with both static and time-varying camera positions and proposed a method for mapping humans gaze fixations onto the 3D shapes with the aim to produce a benchmark of 3D meshes with fixation density maps. Judd et al. [25] collected eye-tracking data from 15 viewers on 1003 images and used this database as training and testing examples to learn a model of saliency based on low,

middle and high-level image features. Kim et al. [26] presented a user study that compares the previous mesh saliency approaches with human eye movements.

In the field of VR, Sitzmann et al. [27] captured and analyzed gaze and head orientation data from a set of users exploring stereoscopic, static omni-directional panoramas, for a total of 1980 head and gaze trajectories for three different viewing conditions. Alexiou et al. [28] performed an eye-tracking experiment to solve the problem of the human viewing behavior in virtual reality environments. And Serrano et al. [29] investigated key relevant questions to understand how well traditional movie editing carries over to VR.

To sum up, eye-tracking is a technique for measuring eye movements to determine where people are looking, what they are looking at, and how long their gaze remains on a particular point. We will use this technique in Section 3 for collecting the necessary data to train our saliency prediction model.

2.2 2D saliency prediction

In the last years saliency prediction has been widely explored. The works by Koch and Ullman [30] and Itti et al. [31] introduced an architecture for saliency detection that extracts multi-scale image features based on color, intensity and orientation. Cornia et al. [32] proposed an architecture made of three main blocks (a feature extraction convolutional neural network, a feature encoding network that weights low and high level feature maps and a prior learning network) that combine features extracted at different levels of a convolutional neural network (CNN). Furthermore, the recent advances in deep learning and, particularly, in convolutional neuronal networks have accomplished to produce more accurate models for saliency prediction [33, 34, 35, 36, 37, 38]. Judd et al. [25] proposed an approach that combines low-level features (color, orientation and intensity) with high-level semantic information (location of faces, cars and text) and showed that this solution considerably improves the ability to predict eye fixations.

In addition, with the ambitious goal of creating an autonomous vehicle, Lasheras-Hernandez et al. [39] proposed a deep learning approach where they resort to a novel convolutional recurrent architecture to learn spatio-temporal features of driving behaviours based on RGB sequences of the environment in front of the vehicle. There are some works that have implemented deep learning models to predict human visual attention in videos as well, such as Chaabouni et al. [40], where the authors attempted to extend CNN approaches in 2D images to detect salient areas in natural video. Moreover, the saliency prediction problem has been extended to 360^{o} content too. For example, SaltiNet [41], ScanGAN360 [42] and the work of Martin et al. [43] are deep neural networks that work with 360^{o} content.

2.3 3D saliency prediction

While 2D saliency prediction has been widely explored, 3D saliency prediction has received much less attention, and it is relevant when users do not see an image of the scene, but a scene with 3D objects that they can interact with. There are several works that have implemented statistical methods and algorithms to predict saliency on 3D meshes.

Lee et al. [8] defined mesh saliency in a scale-dependent manner using a center-surround operator on Gaussian-weighted mean curvatures and compute local mesh saliency as the absolute difference between the Gaussian weighted mean curvature at fine and coarse scales. Leifman et al. [9] proposed an algorithm for detecting regions of interest on surfaces. It looks for regions that are distinct both locally and globally and accounts for the distance to the foci of attention. Song et al. [10] first considered the properties of the log-Laplacian spectrum of the mesh. Then, saliency is captured in the frequency domain by the frequencies which show differences from expected behaviour, and finally, the information about those frequencies is considered in the spatial domain at multiple spatial scales to localise the salient features and give the final salient areas. Tasse et al. [11] proposed a cluster-based approach to point set saliency detection. They first decompose a point set into small clusters, then they evaluate cluster uniqueness and spatial distribution of each cluster, next they combine these values into a cluster saliency function, and finally they use the probabilities of points belonging to each cluster to assign a saliency to each point. There are other works that implemented convolutional neural networks to predict saliency on 3D meshes [12, 2] based on the features extracted by neural networks designed and used for classification problems. Some of the works mentioned above make use of other works [44, 15], in which a set of people have been asked to select the points they think are interesting in some meshes, in order to compare the results of their saliency prediction approaches with what people think that is interesting.

The first work that considered both local contrast and global rarity in order to predict 3D mesh saliency was Wu et al. [45]. They capture local geometric features with various regions. Then they present an efficient patch-based local contrast method based on the multi-scale local descriptor and define global rarity by its specialty to all other vertices. Finally, by the linear combination of the local contrast and the global rarity, the mesh saliency is obtained. Hu et al. [46] also took into account rarity, they proposed a sparsity-enforcing rarity optimization scheme to obtain a compact set of salient regions globally distinct from each other. In addition, mesh saliency can be computed at a fast speed by computing the curvature entropy [47] or the curvature co-occurrence histograms [48] to encode both the global curvature occurrence and the co-occurrence of local distinctive features. The problem of some of the works mentioned is that they rely on handcrafted descriptors [49, 5, 8] and, since their expressive capabilities are limited by the fixed operations that stay the same for meshes of different classes, they do not generalize well.

In the case of saliency prediction for point clouds, there are also a few methods. In the work of Ding et al. [50] they first evaluate the local distinctness of each point based on the difference with its local surroundings. Then the point cloud is decomposed into small clusters and the initial global rarity value of each cluster is computed. Next they use a random walk ranking method to introduce cluster-level global rarity refinement to each point in all the clusters. Finally, they propose an optimization framework to integrate both the local distinctness and the global rarity values to obtain the final saliency detection result of the point cloud. Zheng et al. [51] propose a way of characterizing critical points and segments to build point cloud saliency maps, assigning each point a score reflecting its contribution to the model-recognition loss.

The approach proposed in this project for 3D saliency prediction on point clouds does not use statistical methods or assumptions that establish rarity or curvatures as salient regions. Instead, it is a data-driven technique based on deep learning.

2.4 Deep learning for point clouds

Unlike the statistical methods explained above, deep learning models for saliency prediction on 3D meshes are not so common. Furthermore, at the time of doing this final Master's thesis, only a few works that have used deep learning before for saliency prediction have been found. However, there are several works that have used deep learning for pointclouds, both for object classification and, within a single class of objects, semantic segmentation.

A very common way of working with deep learning and point clouds is voxelization. Usually, researchers voxelize point clouds into 3D grids and then apply a 3D neural network on the volumetric representation [52]. The problems with these methods are that all points within a voxel are assigned the same label, which severely restricts accuracy, and that they are time-consuming and memory-wasting with increasing scale of the point cloud.

Another way of working with point clouds is making use of point-based methods, since they make use of points directly without any transformation. In this context, PointNet [53], a novel efficient and effective type of neural network that consumes point clouds and respects the permutation invariance of points at the input, provides a unified architecture for object classification and part segmentation. Later, since point sets are usually sampled with varying densities, the same authors released PointNet++ [1], a hierarchical neural network that applies PointNet recursively on a nested partitioning of the input point set. This way, the network can learn local features with increasing contextual scales and solves the limitation of PointNet's ability to recognize fine-grained patterns, so that it generalizes better to complex scenes.

More examples of neural networks that work with point clouds are RandLA-Net [54], an efficient neural architecture to directly infer per-point semantics for large-scale point clouds by making use of random point sampling instead of more complex point selection approaches, or EdgeConv [55], a neural network module suitable for CNN-based high-level tasks on point clouds including classification and segmentation.

Finally, respect to saliency prediction on point clouds, Song et al. [2] proposed a network trained in a weakly supervised manner by using the VGG-19[1] model in order to solve the problem of collecting a large amount of vertex-level annotation as saliency ground truth for training the neural networks. Nousias et al. [12] trained a network with saliency maps extracted by fusing local and global spectral characteristics. Liu et al. [3] proposed an attention-embedding strategy for 3D saliency estimation by directly applying the attention embedding scheme to the 3D mesh by making use of PointNet++. This means that regions with very different feature vectors are assumed to be salient, while regions with very similar feature vectors are not. This way, they can train the network without requiring ground truth saliency data. The fact that PointNet++ extracts feature vectors in small subsets of points makes this approach possible.

This Master's thesis presents an approach for 3D mesh saliency prediction based on the neural network from PointNet++ [1] and using real gaze data as ground truth. Despite the fact that this network has been designed for classification and part segmentation, and since it captures local structures induced by the metric space points, the network can learn local features from point clouds even in non-uniformly sampled point sets, and thus can be useful for our problem. For this, it was necessary to adapt the network to our problem modifying the loss

[1] https://keras.io/api/applications/vgg/

function, the output layer of the network and its activation function. These modifications, as well as the architecture of PointNet++, will be explained in Section 4.

Therefore, the main contribution of this work is summarized as follows: (1) a deep learning model for point cloud saliency prediction is proposed, (2) the deep learning model is trained using real saliency data as ground truth, and (3) a comparison between the predicted saliency maps by the model and the ground truth is presented, as well as a comparison between our approach and other saliency prediction approaches for point clouds.

3. Experiment and dataset preparation

This chapter explains the process carried out to collect the gaze data used to train the neural network in order to predict saliency on point clouds. The first step in this process was to get the stimuli, which consisted of a set of 3D meshes (Section 3.1). Once the stimuli were ready, the next step was to set up the virtual reality environment where the experiment was going to be carried out (Section 3.3 and Section 3.4). For this, it was necessary to prepare and configure the hardware and software that was going to be used. When all of this was done and checked, to make sure everything was working fine, a pilot test was performed. Next, the participants were called in to do the experiment (Section 3.5). Finally, when all participants carried out the experiment and their gaze data was collected, that data was processed to prepare the dataset for the neural network (Section 3.6).

3.1 Stimuli

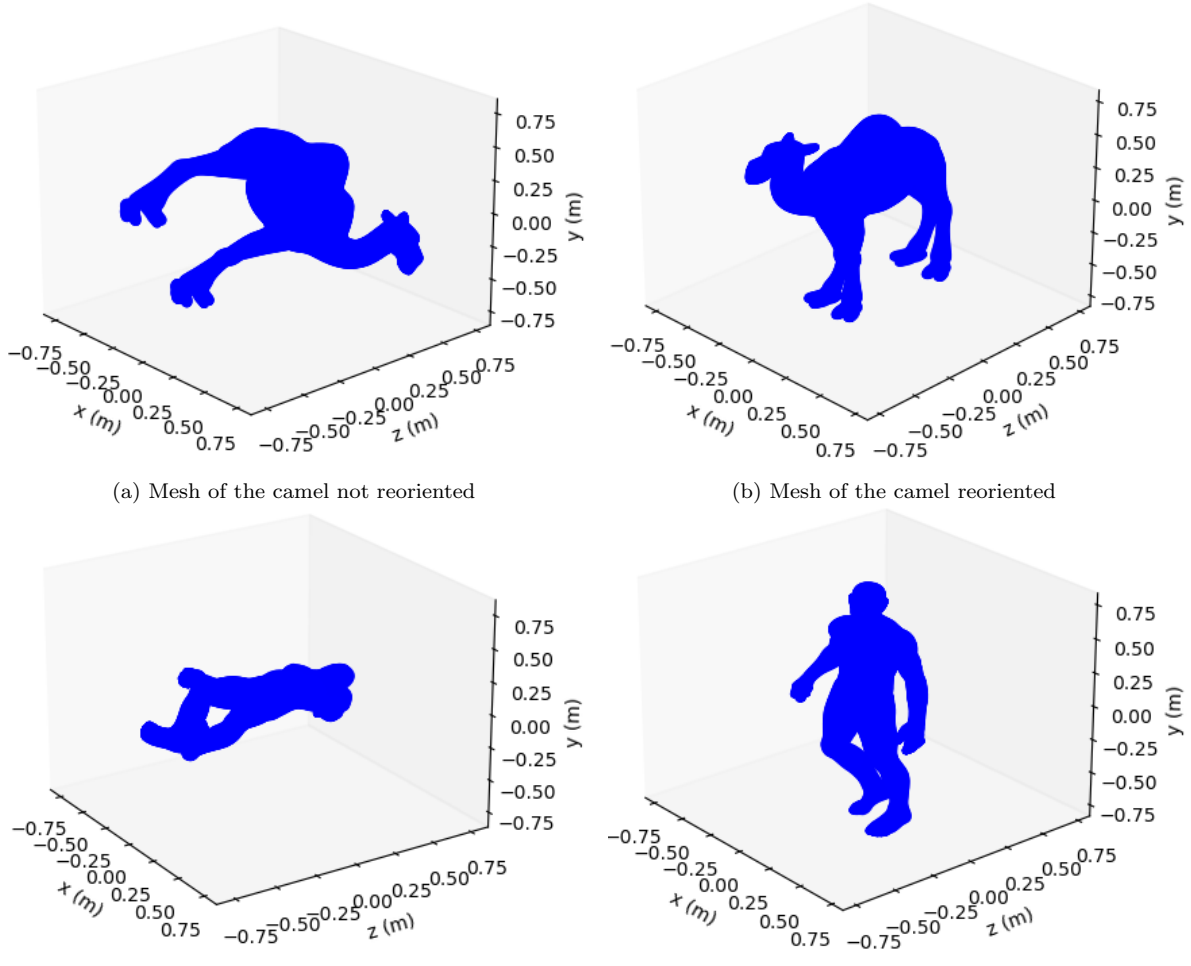
Since the goal of this project is to create a deep learning model capable of predicting saliency on point clouds, the first thing to do in order to carry out the experiment with people was to find and obtain 3D meshes of different classes such as humans, animals and creatures, familiar objects or mechanical parts. The total number of meshes used in this experiment is 60, and they have been taken from different public databases (Aim@Shape[1], TOSCA[2], SHREC 2007[3], Georgia Tech Models Archive[4], FREE3D[5], TurboSquid[6], CGTrader[7]). Some of these meshes have also been used in other works [13, 14, 26] where the authors studied the behavior of human visual attention when looking at different meshes under different conditions: material, point of view, room lighting...

To carry out the experiment, and taking into account that the meshes have been made by different people using different design software, they had to be processed in order to make them be the same size and orientation. Thus, all meshes have been scaled and centered at the origin using the method by Qi et al. [1], paper from which the neural network architecture has been taken and modified to be used in this project. To center the meshes at the origin, the midpoint of the vertices that make up each mesh is computed and then deduced from the original vertices.

[1] http://visionair.ge.imati.cnr.it/ontologies/shapes/
[2] http://tosca.cs.technion.ac.il/book/resources_data.html
[3] http://watertight.ge.imati.cnr.it/
[4] https://www.cc.gatech.edu/projects/large_models/
[5] https://free3d.com/es/
[6] https://www.turbosquid.com/es/Search/3D-Models/free
[7] https://www.cgtrader.com/es/gratis-3d-modelos

On the other hand, to scale each meshes, the coordinates of its vertices are divided by a scale factor, computed as the largest norm of the norms of all its vertices. This way, all meshes are centered at the origin and are forced to be sized to fit a 2-by-2 meter cube centered at the origin (see Appendix E.1 for details).

After this process was done for all the meshes, they were edited to have the same orientation using Blender[8], a free 3D modeling software. This edit was made to prevent the meshes from being upside down. Fig. 3.1 shows a couple of examples of the rotation applied to the meshes in order to reorient them.



(a) Mesh of the camel not reoriented

(b) Mesh of the camel reoriented

(c) Mesh of the human not reoriented

(d) Mesh of the human reoriented

Figure 3.1: Correction of the orientation and rotation of the meshes.

In this project all meshes have been reoriented in the z direction because this orientation matched the orientation of the users in the scene created to carry out the experiment, so when importing the meshes into the virtual reality environment it was not necessary to apply them any rotation. All meshes used in the experiment are shown in Appendix E.2.

[8]https://www.blender.org/

3.2 Participants

A total of 32 participants took part in the experiment, 23 male and 9 female, aged between 21 and 56. They voluntarily took part in the study and provided written consent. The participants were naïve to the purpose of the experiment. From all the participants, 17 people reported having vision problems (such as myopia, hyperopia, astigmatism or presbyopia) and, from these 17 people, only 2 people did not have their vision problems corrected by wearing glasses or contact lenses. 13 participants reported playing video games regularly, and 22 reported having used a virtual reality headset before. From this last group of 22 people, 1 person reported using it very often, 5 people reported using it often and 16 people reported using it occasionally. In addition, also from these 22 participants, 11 people reported having experienced fatigue, eyestrain, dizziness or headaches after using the virtual reality headset. The data collection procedure has been approved by the Data Protection Unit of the University of Zaragoza and by the CEICA (Ethics and Research Committee of Aragon, by its acronym in Spanish).

3.3 Hardware and data collection

The experiment has been carried out in VR. This decision was made because of three main reasons: (1) it offers more control over the viewing conditions, (2) it provides stereo viewing and (3) it avoids possible distractions from the real world. In addition, the research group has the necessary equipment available for it, the software used to design scenes in VR is intuitive and has very useful functions for collecting data already implemented and ready to be used, and also the eye tracker equipped in the virtual reality headset has a plugin specifically designed to directly collect data when it is used along with that software.

The equipment used to perform the experiment consists of the HTC Vive Pro[9] virtual reality headset and the Pupil Labs[10] eye tracker. As mentioned above, since the Pupil Labs plugin to extract data from the eye tracker is designed and programmed for the Unity[11] game engine, this has been the tool utilised to design the scene and carry out the experiment.

The headset has a nominal field of view of 110°, with a resolution of 1440×1600 pixels per eye (2880x1600 pixels combined), and a frame rate of 90 frames per second. In addition, to get a better approximation of the position of the headset in space and reduce the probability of tracking failure, two tracking sensors (HTC Vive stations) have been used. The headset device is also equipped with the binocular eye tracker from Pupil Labs.

To obtain the point that the user is fixating on the scene, the position and orientation of the head at each instant of time are needed. The tracking sensors provide this information to the software, and it is then used and processed by the Pupil Labs plugin to provide the rest of the information related to the gaze data: 3D collision point with the mesh (in local and world coordinates), distance between the position of the head and the collision point, gaze confidence... All useful information provided by the eye tracker is stored in a *csv* file after the user has seen each mesh, which means that there will be a *csv* file per mesh and per participant.

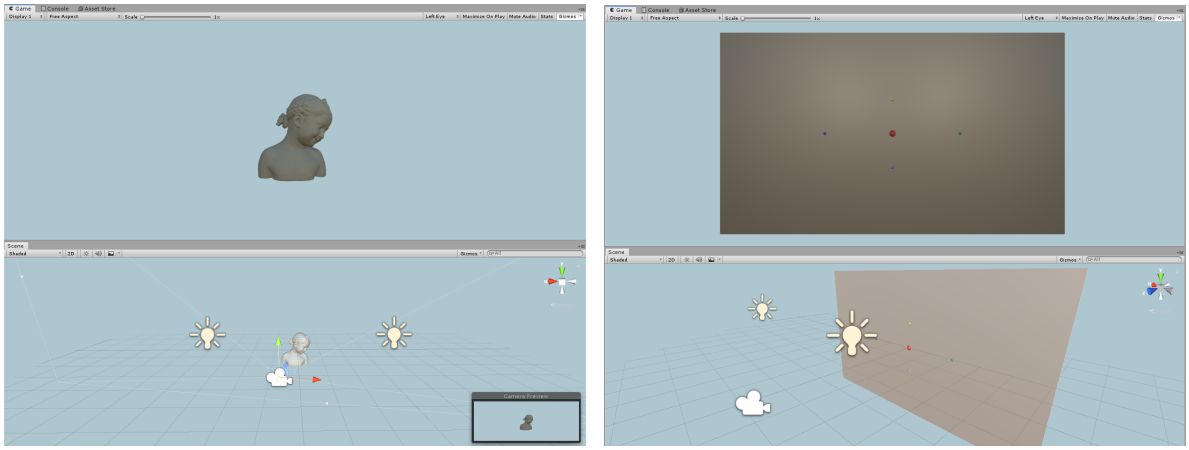
[9] https://www.vive.com/us/product/vive-pro-full-kit/
[10] https://pupil-labs.com/
[11] https://unity.com/es

3.4 Scene

When an experiment like this is going to be carried out, in order to prevent the participants from being distracted and keep their attention on the scene, it is necessary that the scene designed for it meets a series of features. For example, colours can influence the users' attention, as well as the illumination of the scene and the shadows generated by the illumination.

To prevent users from being distracted because of colours, the environment created consists of a big light blue sphere, within which the users are located, and a neutral grey colour for the meshes. On the other hand, to prevent users from being distracted because of the illumination and shadows, two light sources have been used, located above the users and slightly to the left and right, since this combination was the one that best fit the mentioned features. Fig. 3.2a shows the scene designed for this experiment.

Finally, to check if the eye tracker calibration has been successful, an intermediate scene was designed. This scene consists of a brown wall and five different coloured balls, and appears every time a mesh view time expires. After calibration, participants are asked to look at the different balls, and, if the eye-tracking works correctly, the experiment begins. Fig. 3.2b shows the calibration scene.



(a) Experiment scene (b) Calibration checking scene

Figure 3.2: Scenes designed for the experiment. For each scene, we show the point of view of the participant (top part of the images) as well as a point of view that allows to see both the scene and the position of the participant (camera icon) in the scene (bottom of the images).

3.5 Procedure

To ensure the safety of participants, in order to avoid fatigue, dizziness, eyestrain or other symptoms that may appear when using virtual reality headsets, the experiment has been divided into two sessions. In each session, half of the meshes are seen and the process to carry out the experiment is practically the same in both sessions. The meshes appear in random order for each participant and session.

Initially, participants were explained what the experiment consisted of and were asked to

sign the informed consent (see Appendix A), a document required by the government of Aragón when data is to be collected in experiments with people. After this, participants must complete a brief demographic questionnaire (see Appendix B) and a brief sickness questionnaire to know if they have any vision-related symptoms prior to the experiment (see Appendix C). When this is done, the experiment begins.

To make all participants see the meshes from the same point of view and from the same distance, they remain seated during the experiment in a chair located approximately one meter from the tracking sensors (see Fig. 3.3). The participant sits down, puts on the headset and adjusts it to the head. The first step is the eye-tracking calibration process, which is repeated if the accuracy is not precise enough. To see if the accuracy is precise, the participant is asked to look at the different balls. If the accuracy is valid, the S key on the keyboard is pressed, and when the participant looks at the red ball while the S key is pressed, the first mesh appears.

Figure 3.3: Experiment setup. The participant sits in a non-swivel chair, wearing the virtual reality headset and approximately one meter from the tracking sensors.

In order for the participant to see the entire mesh, each mesh performs a complete rotation on itself. This rotation has been set to last 22 seconds, so that it is fast enough to keep the participant's attention and avoid loss of concentration, but also slow enough so that the participant has time to see all the details of the meshes. When the rotation is finished, the collected data is stored in a *csv* file and the intermediate scene to check if calibration is working properly appears. If the calibration is still correct, the process of pressing the S key until the participant looks at the red ball to show the next mesh is repeated, and so on. After finishing the session, participants are asked to complete the sickness questionnaire again to see how the experiment influenced possible vision-related symptoms. In order to rest and give these possible symptoms time to disappear, the participants must take a break of, at least, 30 minutes before carrying out the second session.

In the second session the procedure is almost the same. First, participants complete the sickness questionnaire prior to the experiment, as before. Then they put on the headset and carry out the experiment with the other half of the meshes. The calibration and mesh appearances

processes are the same as before. At the end of the session they complete again the sickness questionnaire, subsequent to the experiment, and a presence questionnaire (see Appendix D) to measure if presence and engagement influenced the perception of the animation. Finally they are thanked for their participation.

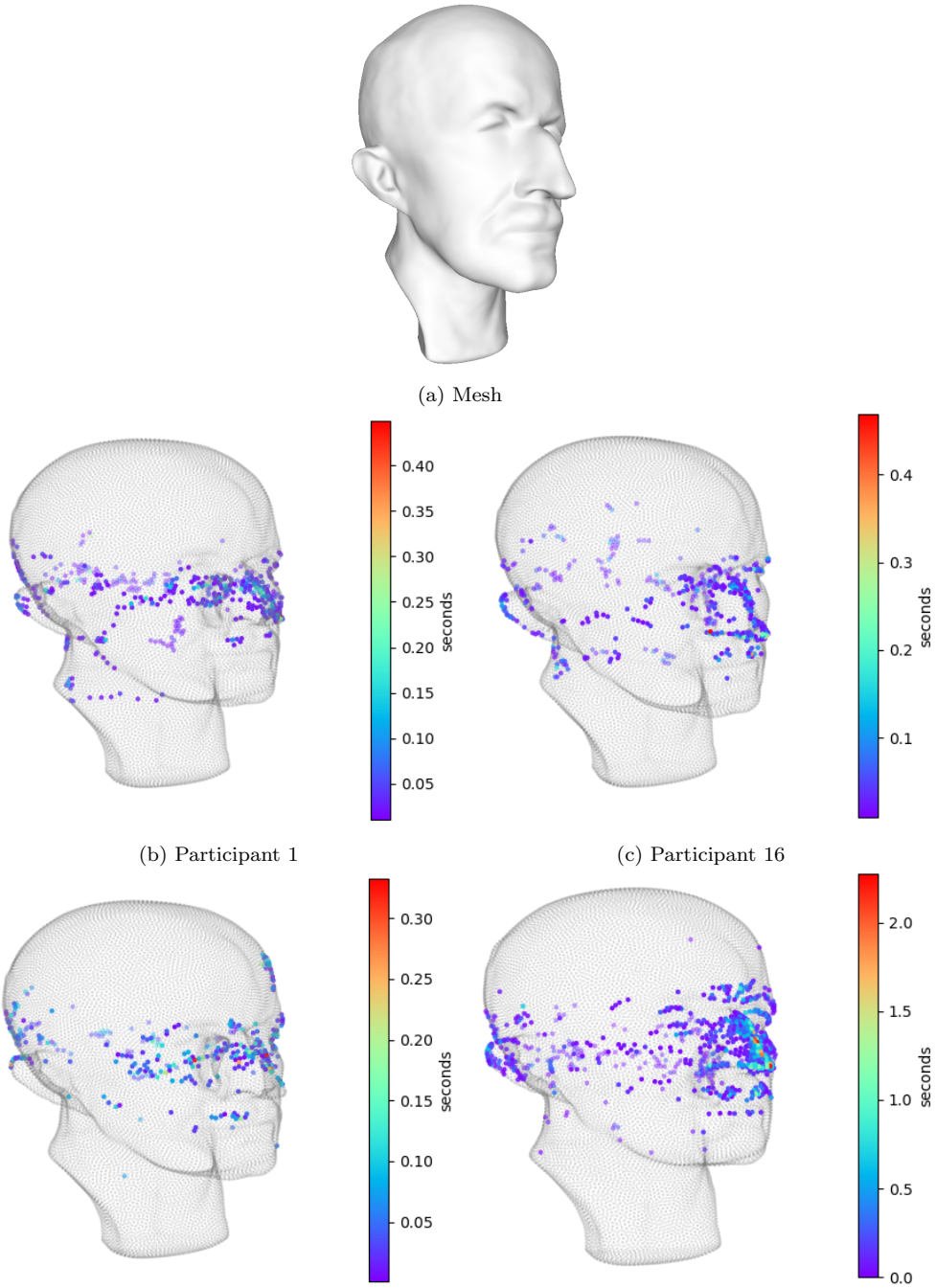
3.6 Data processing

As mentioned in Section 3.3, the Pupil Labs plugin provides information captured by the eye tracker in real time such as head position, head orientation, gaze direction, collision point with elements in the scene, confidence of each gaze point, etc. All this information is stored in different *csv* files so that, once the experiments are finished, they can be processed to create the fixation maps of each mesh. Fixation maps are the time each vertex has been fixed. The data recorded in these *csv* files were the timestamps, along with the rest of the information provided by the eye tracker. When the experiments were finished, all files were processed to create the fixation maps of each mesh for each participant. Because 32 people participated in the experiment, and since each participant saw 60 meshes, a total of 1,920 fixation maps were created.

To create each fixation map, the *csv* files are processed individually. Initially, a vector of the same size as the number of vertices of the mesh is created for each *csv* file. Then, for each recorded data, it is checked if the collision point corresponding to that timestamp has been with the mesh. If this is the case, a fixation time of $\Delta t = timestamp_{i+1} - timestamp_i$ is added to the fixation map at the index corresponding to the vertex of collision with the mesh. Otherwise, this recorded datum is skipped. This process is repeated for all the *csv* files, so that in the end there are 1,920 fixation maps, one per mesh and participant. The last step is to aggregate all fixation maps per mesh, so that in the end there is only one fixation map per mesh. The aggregated fixation maps were normalized between 0 and 1 as shown in Eq. 3.1. As an example, Fig. 3.4 shows the Max Planck mesh, along with the mesh fixation maps of three randomly selected participants, as well as the aggregated fixation map of all participants.

$$fixmap = \frac{fixmap - \min(fixmap)}{\max(fixmap) - \min(fixmap)} \tag{3.1}$$

To end the dataset preparation, it is necessary to convert the meshes to the format **.xyz*, since the input to the neural network used in this project is the 3D coordinates of the vertices that make up the mesh, as well as the normals of those vertices. This way, the dataset is ready. On the one hand there are the inputs of the neural network, which are the vertices and their normals, and on the other hand there are the outputs, which are the fixation maps corresponding to each vertex and normal. Different tests have been carried out to train the neural network, and the results obtained will be discussed in Section 5.



(a) Mesh

(b) Participant 1

(c) Participant 16

(d) Participant 30

(e) Aggregated

Figure 3.4: Example of fixation maps. Image a) is the original mesh. Images b), c) and d) are the fixation maps of 3 participants. Image e) is the aggregated fixation map of all participants for this mesh. Grey points are non-fixated points and coloured points are points that have been fixated. Warmer colors represent the salient parts of the mesh.

4. Model for 3D mesh saliency prediction

As explained at the end of Section 2.4, the deep learning model utilised in this project in order to predict saliency on point clouds has been PointNet++ [1]. The general idea of PointNet++ is defined as follows: partition a point cloud into local regions, extract local features by capturing fine geometric structures from small neighborhoods, group those features into larger units, and finally process them to produce higher-level features. This process is repeated until features of the entire point cloud are obtained. In this chapter the hierarchical architecture of PointNet++ is explained, as well as the modifications made to adapt it to the saliency prediction problem.

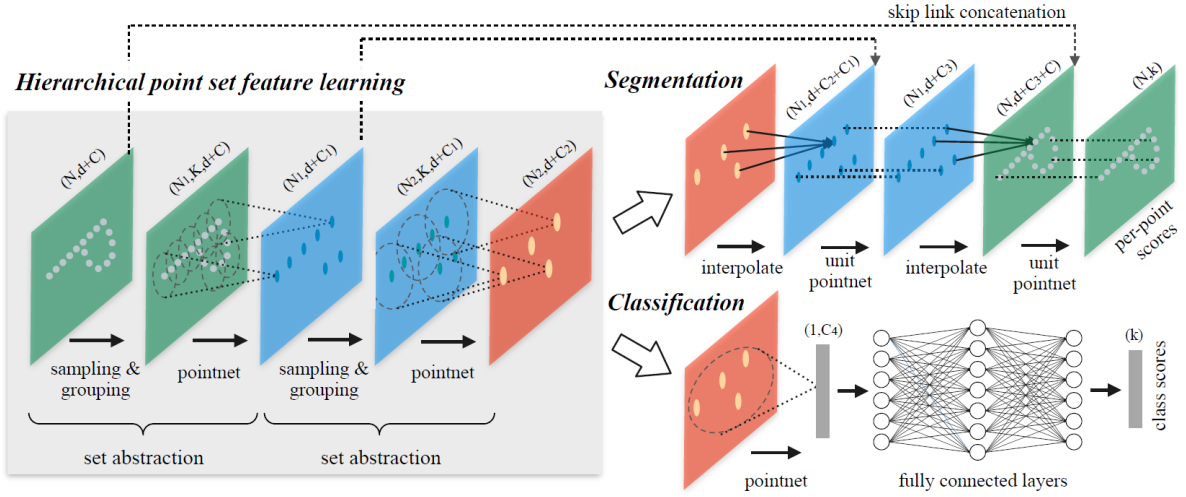


Figure 4.1: Hierarchical feature learning architecture of PointNet++ using points in 2D Euclidean space as an example. Image extracted from the original PointNet++ work [1].

PointNet++ is a neural network designed for classification and part segmentation. It is capable of classifying point clouds according to the object class they belong, and also, within a single class of objects, it is capable of classifying its parts. To train these type of networks, there are some available datasets that contain point clouds with the corresponding labels for each point such as ModelNet40 [56], ShapeNet [57], S3DIS [58] or ScanNet [59]. For our problem, the PointNet++ variant for part segmentation was used (top right variant in Fig. 4.1), since point features need to be obtained for all points in the point clouds.

4.1 Architecture

The goal of PointNet++ is to learn set functions f that take sets of points as the input and produce information of semantic interest. The network is composed by a number of *set abstraction* levels and a set of *feature propagation* levels. *Set abstraction* levels perform the task of processing sets of points, abstracting feature vectors from them and producing a new set with fewer elements (see grey box on the left in Fig. 4.1). On the other hand, *feature propagation* levels propagate point features obtained from the *set abstraction* levels (see segmentation variant in Fig. 4.1).

4.1.1 Set abstraction levels

Each *set abstraction* level is made of three layers: a sampling layer, a grouping layer and a PointNet layer (grey box in Fig. 4.1). They take as input a $N \times (d + C)$ matrix, where N is the number of points, d is the dimension of the coordinates and C is the point feature dimension. As output, they return a $N' \times (d + C')$ matrix, where N' is the number of subsampled points, d is the dimension of the coordinates and C' is the dimension of the new feature vectors summarizing local context. Next, the layers that make up the *set abstraction* levels are detailed.

Sampling layer. This layer selects a subset of points from the input points, which define the centroids of the local regions. These centroids are chosen by using the Farthest Point Sampling algorithm, which iteratively samples data from a point cloud. It starts from a single random point and, at each iteration, selects the farthest point from the selected subset of points.

Grouping layer. The grouping layer builds sets of local regions by finding neighbour points around the centroids defined in the sampling layer. The input to this layer is a set of points of size $N \times (d + C)$ and the coordinates of a set of centroids of size $N' \times d$. The output are groups of point sets of size $N' \times K \times (d + C)$. Each of these groups corresponds to a local region, being K the number of points in the neighbourhood of the centroids. To find the neighbours of the centroids, the PointNet++ implementation makes use of the Ball Query algorithm, a method that finds all points that are within a radius from the query point.

PointNet layer. To encode local region patterns into feature vectors, a mini-PointNet is used. The input to this layer are N' local regions of points with size $N' \times K \times (d + C)$. The local regions in the output are abstracted by their centroids and local features that encode the neighbourhoods of the centroids. The size of the output data in this layer is $N' \times (d + C')$.

In addition to this, it can happen that point sets have different point densities in different areas. The problems that arise from this situation are that, in case a model is trained with dense data, it may not generalize well to sparsely sampled regions, and vice versa, models trained for sparse point clouds may not recognize high frequency details in local structures. To solve the problem, the authors of PointNet++ proposed density-adaptive PointNet layers, a type of layers that are capable of learning to combine features from regions of different scales when the input density changes. At each *set abstraction* level, the density-adaptive PointNet layers extract multiple scales of local patterns and intelligently combine them according to local point densities. The type of density-adaptive layer proposed is explained in the next paragraph.

Multi-resolution grouping. The main idea of this density-adaptive layer is that features of a region at some level, L_i, is a concatenation of two vectors. One of them is obtained by summarizing the features at each subregion from the lower level L_{i-1} using the *set abstraction* level (darkest vector on the left in Fig. 4.2), and the other one is the feature obtained by directly processing all raw points in the local region using a single PointNet (lightest vector on the right in Fig. 4.2).

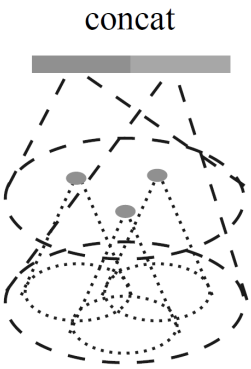


Figure 4.2: Density-adaptive PointNet layer vector concatenation. Image from PointNet++ [1].

Therefore, if the density of a local region is low, the second vector will be more reliable than the first one and should be weighted higher. On the contrary, if the density of a local region is high, the first vector will provide information of finer details.

4.1.2 Feature propagation levels

As mentioned in the introduction of this chapter, for the semantic point labeling task it is necessary to obtain point features for all the original points. For that, the solution proposed by the authors of PointNet++ was to propagate features from subsampled points to the original points, adopting a hierarchical propagation strategy with distance-based interpolation and jump links between levels (as shown on top right in Fig. 4.1).

In *feature propagation* levels, point features are propagated from $N_l \times (d+C)$ points to N_{l-1} points, where N_{l-1} and N_l are the point set sizes of the input and output of the *set abstraction* level l. By interpolating feature values of N_l points at coordinates of the N_{l-1} points, the feature propagation is achieved. Next, the features interpolated are concatenated with skip linked point features from the *set abstraction* level. Finally, the concatenated features are passed through a unit PointNet, a network similar to a one-by-one convolution. This process is repeated until the features had been propagated to the original set of points. In the end, what the network returns is a vector with the probabilities of the points to belong to each of the parts in which the input point cloud has been segmented.

4.2 Model modifications

PointNet++ is a neural network designed for a classification problem. Therefore, it has been necessary to make some modifications to the original network in order to adapt it to the saliency prediction problem. These modifications have been three and are detailed below.

Output layer. In a classification problem, the output layer has the same number of neurons as possible classes for the input data. For the part segmentation problem, depending on which mesh is used to train the network, that number may not be the same for all meshes. As an example, if the network is trained to learn the parts that make up an airplane, the number of neurons in the output layer could be 4: the body, the wings, the engines, and the tail. Thus, the size of the output layer would be 4, corresponding to the four probabilities predicted by

the network that a point in the point cloud belongs to those parts. In the saliency prediction problem, since each point is assigned an only continuous value, corresponding to its fixation map, the output layer has only one neuron. Ideally, the value predicted by the network at the output layer in this neuron should be as similar to the ground truth fixation map as possible.

Output layer activation function. The authors from PointNet++ do not use any activation function at the output layer. They directly take the values predicted by this layer and pass them through the function that automatically computes the loss value. This function (explained in the next paragraph) is already implemented in the Tensorflow API *tf.nn*[1] and it already applies the supposed activation function on the predicted values before computing the loss. In this project, the values predicted for each point by the network represent the probability that a human will fix that point. Since this values are 0 or larger than 0, the output layer activation function used has been the $ReLU$[2] function (see Eq. 4.1).

$$fixmap = ReLU(fixmap) = \max(0, fixmap) \tag{4.1}$$

Loss function. The loss function used by the authors in the original PointNet++ implementation is the *sparse softmax cross entropy with logits*[3]. This loss function is utilised in discrete classification tasks where classes are mutually exclusive (i.e. each input data belongs to only one class) and it measures the probability error. First, this function applies the softmax activation function to the values predicted by the network at the output layer and then computes the cross-entropy loss between the actual and the predicted labels. This is the reason why the authors do not use an activation function in the original implementation of PointNet++ at the output layer. For example, in the CIFAR-10[4] dataset, each image is tagged with only one tag and cannot have another tag. In the saliency prediction problem, the points in the point clouds are not tagged with labels, but have assigned a continuous value, which corresponds to the time that they have been fixated in total by all the participants who carried out the experiment. Since this is a regression problem, the loss function that best fitted the task is the *mean squared error* (MSE), a loss function that measures the mean squared difference between the predicted values and the actual values. The mathematical equation that defines this loss function is shown in Eq. 4.2, where y_i represents the actual fixation map of a vertex of the mesh, $\hat{y}_i$ represents the predicted fixation map for that vertex, and N is the number of vertices over which the error has been computed.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{4.2}$$

4.3 Training details

To train the model, since most of the meshes had 20,000 vertices, all the meshes with their corresponding fixation maps have been normalized to this number of vertices. Meshes with more than 20,000 vertices had the corresponding number of vertices removed to leave them with this

[1] https://www.tensorflow.org/api_docs/python/tf/nn
[2] https://pytorch.org/docs/stable/generated/torch.nn.ReLU.html
[3] https://www.tensorflow.org/api_docs/python/tf/nn/sparse_softmax_cross_entropy_with_logits
[4] http://www.cs.toronto.edu/ kriz/cifar.html

number of vertices. In order not to remove all vertices from a specific part of the mesh and break its geometry, the vertices were randomly shuffled and then the indices of the vertices to be removed were also randomly selected. Meshes with less than 20,000 had the corresponding number of vertices added to establish them with this number of vertices. To carry out this addition task, the Blender[5] software was used. Initially a copy of the meshes is created and imported to Blender. Next, their faces are split to multiply the number of vertices. Then, this new meshes are exported. In order not to add repeated vertices, the vertices of the new meshes that are already in the original meshes are removed. From the remaining points in the new meshes, the number of vertices needed to set the original meshes to 20,000 vertices is randomly selected, and those vertices are added to the original meshes. Finally, the fixation maps for each of these new vertices are assigned by finding their nearest neighbours in the original mesh.

With respect to the hyperparameters, they are configurable parameters by the user whose values control the learning process and determine the values of model parameters that a neural network ends up learning. Some of the most typical hyperparameters are the number of epochs, the batch size, the learning rate and optimisers. The epochs indicate the number of passes of the entire training dataset that the network has completed. The batch size refers to the number of training samples utilized in one iteration. The learning rate determines the step size at each iteration while moving towards a minimum of a loss function and represents the speed at which a neural network learns. Finally, optimisers are algorithms used to change the attributes of a deep learning model such as weights and learning rate in order to reduce the losses.

In this project, the training process took around 3 hours on a NVIDIA RTX 3060 GPU. The number of epochs of the training was set to 300, the batch size was 1, the optimiser used was the Adam [60] optimiser, with a learning rate $lr = 10^{-3}$ and a weight decay [61] $w_d = 10^{-4}$. Also, the learning rate scheduler StepLR[6] was added to the optimiser in the training process. The learning rate schedulers are predefined frameworks that decay the initial learning rate with some multiplicative factor every N epochs and can give better training performance and make the model converge faster.

[5]https://www.blender.org/

[6]https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.StepLR.html

5. Results and evaluation

This section provides an analysis of the performance of the proposed model for saliency prediction on point clouds. Initially, in Section 5.1, the metrics utilised for measuring the performance of the model are detailed. Next, the results obtained with the proposed model are shown in Section 5.2. Then, a comparison with previous state-of-the-art works is presented in Section 5.3. Finally, Section 5.4 offers an ablation study that validates the decisions made for the implementation of the model.

5.1 Metrics

In order to evaluate the performance of the developed model and provide a meaningful comparison with other state-of-the-art works, three different metrics commonly used in regression problems were chosen. To compute these metrics, both the ground truth and the predicted fixation maps have been normalized between 0 and 1, where a value of 1 would mean a high probability for the vertex to be fixated and 0 would mean that the probability is null. The metrics used to measure the difference between the predicted saliency map P and its ground truth G, obtained from the real gaze data collected in the experiment, are the following:

- Mean Squared Error (MSE): measures the average squared difference between the predicted values and the actual values. In the saliency prediction problem, this metric computes the error between the values of the ground truth fixation maps and those predicted by the model. It is computed as shown in Eq 4.2.

- Mean Absolute Error (MAE): measures the average difference between the predicted values and the actual values. Unlike the MSE, in this metric all individual differences have equal weight. Eq 5.1 shows how the mean absolute error is computed. In the same way as MSE, y_i is the actual fixation map of a vertex of the mesh, $\hat{y}_i$ is the estimated fixation map for that vertex, and N is the number of vertices over which the error has been computed.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{5.1}$$

- Pearson's Correlation Coefficient (CC): it is a statistical method used to measure the correlation, association or dependence between two continuous variables. In the particular problem of saliency prediction, this coefficient measures the linear relationship between

the two saliency maps (P and G). Its values can range from -1 to 1, where -1 indicates a perfect negative relationship, 1 indicates a perfect positive relationship and 0 indicates no relationship. The CC is computed as shown in Eq. 5.2, where cov is the covariance, σ_P is the standard deviation of P and σ_G is the standard deviation of G.

$$CC(P, G) = \frac{cov(P, G)}{\sigma_P \cdot \sigma_G} \tag{5.2}$$

The values of these metrics presented in the ablation study represent the mean and the standard deviation of the measurements obtained for the 4 test point clouds. The mean values and the standard deviations are computed by evaluating the predicted fixation maps for each point cloud against their corresponding ground truths, and then the obtained values are averaged to obtain the final model performance.

5.2 Results

The division of the dataset with which the results of the proposed model have been obtained is as follows: 2 point clouds discarded because of some artifacts in the 3D model files, 48 point clouds in the training process were used for the training loop and 6 point clouds for the validation loop, and 4 point clouds were separated to test the model once the training process was finished.

The 4 point clouds separated for testing the model were selected in order to see its performance on different classes of objects. Thus, it is possible to see how the model is capable of predicting the most salient parts of the meshes, or the regions of the meshes that are most likely to attract human visual attention (see Fig. 5.1).

- In the case of the car, the ground truth shows that the parts that most caught the attention of the participants in the experiment were the front and rear of the car, as well as the wheels and their surroundings. In the prediction, it can be seen how the model is able to detect these salient regions, focusing the prediction mainly on the front and rear of the car, but also on the wheels and even on the side mirrors.
- The case of the camel shows how the model is able to predict that the most salient part of the mesh is the head, what matches with the ground truth and the experiment carried out by Lavoué et al. [13].
- For the Jessi mesh, again, the model is able to predict that her face is the part that a human would be most likely to focus. The predictions agree reasonably well with both the ground truth and the experiment carried out by Lavoué et al. [13].
- Finally, the watchtower is a mesh with more scattered attention than the other three. This makes sense, since it is a quite uniform object with no relevant salient parts, and it is because of this reason that the model is more hesitant in predicting which parts are more salient and spreads the predicted saliency values more throughout the point cloud.

Fig. 5.1 shows the saliency results obtained when making predictions on the test point clouds. The warmer colors represent the salient parts of the objects, i.e. the parts of the objects that

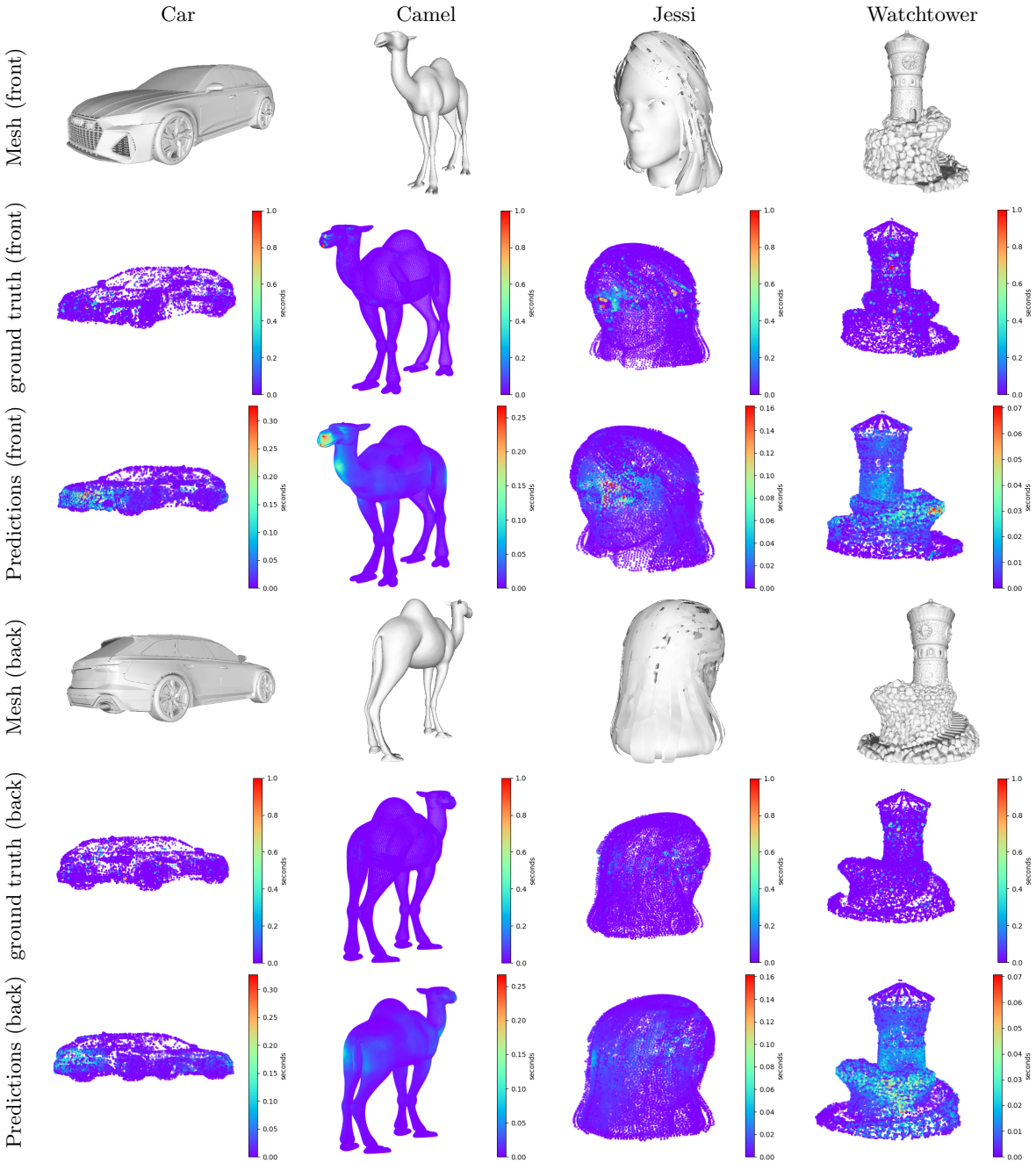


Figure 5.1: Saliency prediction results. The first three rows show the original mesh, the saliency of the ground truth, and the saliency predicted by our model seen from the front view, respectively. The last three rows show the original mesh, the saliency of the ground truth, and the saliency predicted by our model seen from the back view, respectively. Saliency is represented as a color map blended with the point cloud, where warmer colours correspond to more salient areas.

are most likely to attract human attention. As can be seen, the proposed approach can detect the parts of the point clouds that are visually significant. However, the values of the fixation maps predicted by the model for the test point clouds are much lower than the original ones. The reason why this happens is because most of the ground truth fixation maps have a value of

0 or a value very close to 0, so the model understands that the best way to get its predictions closer to the ground truth is predicting very low values.

5.3 Comparison with previous work

The approach proposed in this project has been compared with other state-of-the-art works that also predicted saliency on 3D meshes. On the one hand, the selected works that predict saliency using statistical methods have been Tao et al. [62], Song et al. [10], and Tasse et al. [11]. On the other hand, the selected works that predict saliency using deep learning have been Song et al. [2] and Liu et al. [3]. All these works used the ground truth dataset provided by Chen et al. [15] to compare their results. The ground truth provided by Chen et al. has been created doing an experiment in which the authors asked people to manually select points of the meshes that are most likely to be selected by the other participants as interesting. After the experiment, they performed a post-processing to create the saliency maps (more details in their work [15]). So, in order to compare our results with theirs, saliency predictions on the meshes provided by Chen et al. have been made.

First, the results are compared qualitatively with the works that predict saliency using statistical methods [62, 10, 11] along with the ground truth. These results are shown in Fig. 5.2. It can be seen that our approach tends to generalize reasonably well. The clearest examples are the vase at the first column, the chair, the bacterium and the bear. In these cases our approach is able to detect the parts marked as salient in the ground truth, while the other approaches tend to fail due to the assumption that the parts with steep curvatures and corners are salient. The cases of the human and the hand are the least close to the ground truth. In the human, the rest of the works tend to predict the man's head and limbs as salient since they are the parts with strongest curvatures, what in this case matches the ground truth, while our approach gets to detect the man's chest and the beginning of the legs, but fails to detect the head. In the case of the hand, our approach and the approaches of Tao et al. [62] and Song et al. [10] are very similar, the three predict the corners of the bent fingers and the surface of the outstretched fingers as salient and leave the palm as not salient (as the ground truth), while the predictions of Tasse et al. [11] on the hand focus mainly on the fingertips and also predicts the palm as salient, which does not match the ground truth. The cases of the vase in the fifth column and the pig are quite similar for all approaches, the attention on the vase is mainly focused on the rim and handles, and on the pig, even though our approach does not detect the head, the attention is focused on the hooves and the tail, coinciding with the rest of the approaches and the ground truth. Finally, the poorest prediction for all approaches has been the glasses. In this case, as the glasses are a difficult mesh to make predictions, none of the approaches matches the ground truth, but they all match in the rims.

Next, our approach is compared qualitatively with the works that utilised deep learning to make saliency predictions on meshes [2, 3]. As mentioned before, these approaches take advantage of the features extracted within the hidden layers of existing networks designed for classification and assume that if those features are different, that part of the mesh is salient, so they assign a larger weight to those vertices. The expected results for this works are the same than in the previous ones, that is, areas with high-frequency details and strong curvatures will tend to be predicted as salient, while flat areas will not.

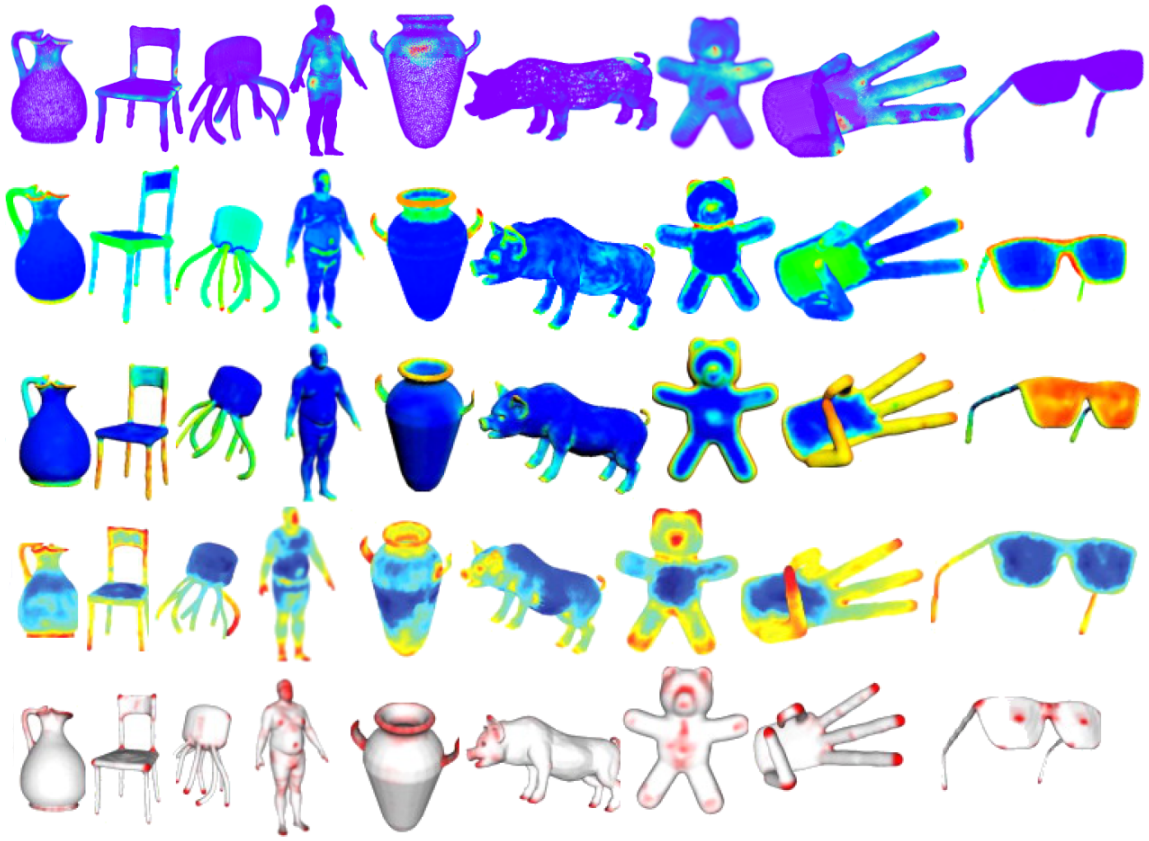


Figure 5.2: Comparison of saliency prediction of our approach with statistical methods. Rows from top to bottom: our approach, Tasse et al. [11], Tao et al. [62], Song et al. [10], ground truth from Chen et al. [15]. Source of the image: Tasse et al. [11]. Warmer colours correspond to more salient areas.

In Fig. 5.3 the comparison between our results and the results from Song et al. [2], along with the ground truth are compared. It is possible to see that our approach again tends to predict reasonably well the most interesting parts of each mesh. In the case of the girl, both our approach and Song's are able to detect that the most salient areas are the head, the chest, the hands and the feet. However, comparing the two approaches with the ground truth, our approach fails to detect the face and the knees, while Song's approach detects all body as salient, which does not match the ground truth. Respect to the armadillo, our approach is able to detect that the most salient areas are the face, the ears, the front knee and the front foot fingertips, but it does not detect the hands and also detects the entire body as salient, which does not match the ground truth. On the other hand, Song's approach, thanks to the feature vectors assumption (different feature vectors correspond to salient parts), detects reasonably well the salient parts of the armadillo. Predictions on the vase are similar for both approaches, they match on the handles and on the top part, but, respect to the body of the vase, because of the feature vectors assumption, our approach matches better with the ground truth than Song's. The rest of the cases, the dolphin, the bird, the hand and the teapot are also very similar, both approaches detect reasonably well the salient parts. Nevertheless, in the hand and the teapot, because of the feature vectors assumption, Song's approach predicts as salient all fingers and the body of the teapot, which does not match the ground truth.

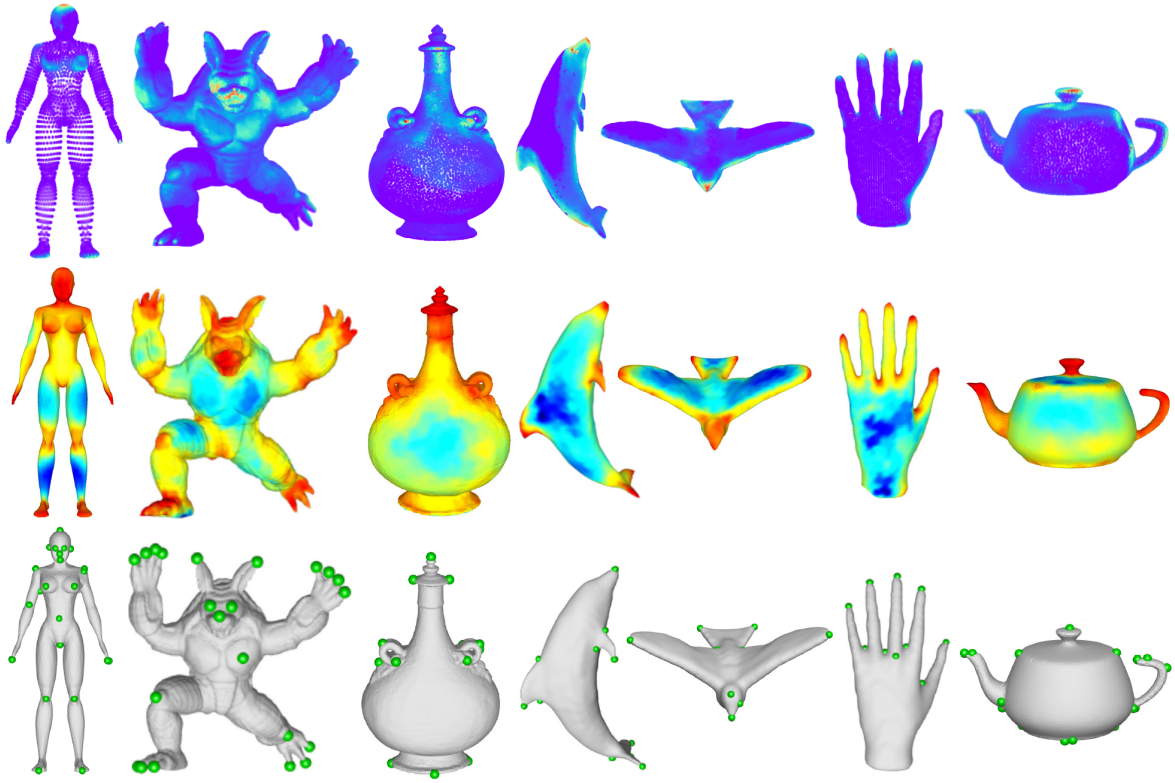


Figure 5.3: Comparison of saliency prediction with the approach of Song et al. [2]. Rows from top to bottom: our approach, Song et al. [2] and the ground truth from Chen et al. [15]. Warmer colours correspond to more salient areas. Source of the image: Song et al. [2].

Finally, the comparison between our approach and the approach of Liu et al. [3] is shown in Fig. 5.4. The results are similar to the previous analysis, for some meshes our approach works better and for other meshes it works worse. The case of the dog is quite clear for the two approaches, since for both of them the most salient part is the head. However, because Liu's approach works with the same assumption than Song's, it also detects the hooves and the tail as salient too, since they are parts of strong curvatures and high-frequency details. For the vase in the second column both approaches detect the handles and the rim as salient, but our approach also detects the part of the body under the rim as salient, which does not match the ground truth, and Liu's approach detects as salient the base of the vase, which does not match the ground truth either. Respect to the vase in the third column, Liu's approach focuses mainly on the top of the vase, which matches the ground truth, but it does not detect the handles (while our approach does) in addition to the saliency predicted on the base, which does not match the ground truth. On the bust, Liu's approach focuses mainly on the nose, chin and neck, and ignores the eyes and mouth, which does not match the ground truth. Meanwhile, our approach focuses on the eyes, the nose, the mouth and the chin, so in this case our approach performs better. In the case of the armadillo, because of the feature vectors assumption, Liu's approach detects the interesting parts reasonably well, while ours does not perform so well. The dolphin predictions are very similar, the most salient parts are detected, but our approach further extends the predicted saliency over the mesh, so Liu's approach matches better the ground truth. The bear is similar for the two approaches, both detect the end of the legs and

arms as salient, as well as the chest, what matches the ground truth. However, Liu's approach does not detect the bear's face as salient, which does not match the ground truth, whereas ours does. On the other hand, our approach detects ears with a low level of saliency, while Liu's detects them reasonably well. Finally, on the person, Liu's approach detects the feet and hands very well due to the feature vectors assumption, but places the face in a second level, in addition to not predicting the saliency of the chest.

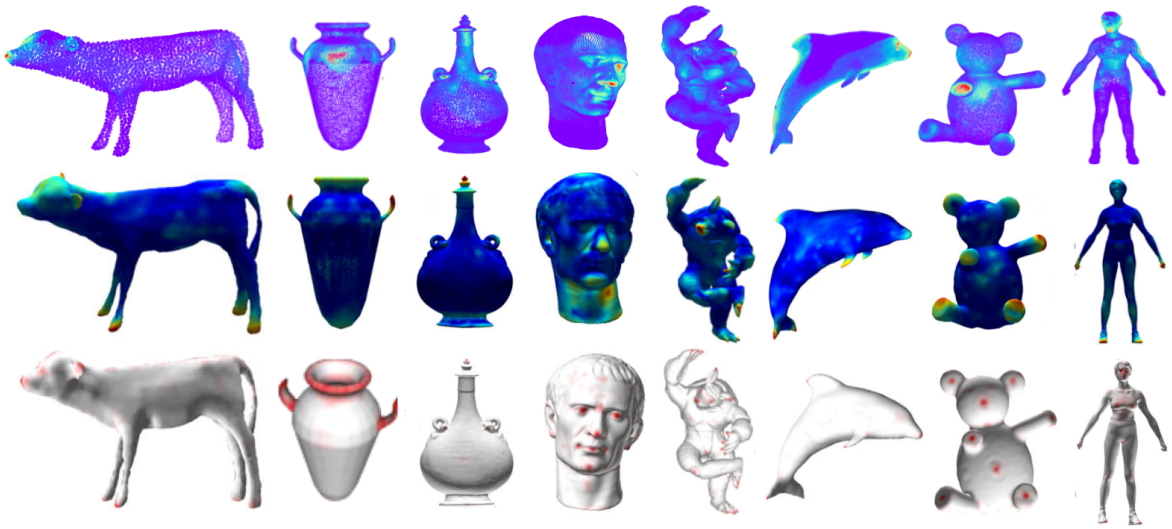


Figure 5.4: Comparison of saliency prediction with the approach of Liu et al. [3]. Rows from top to bottom: our approach, Liu et al. [3] and the ground truth from Chen et al. [15]. Warmer colours correspond to more salient areas. Source of the image: Liu et al. [3].

The authors of these works do not provide the files with their predicted saliency maps, and since we tried to reproduce the code provided by the different authors but it did not work, the comparison of our approach with these works has been qualitatively.

5.4 Ablation study

In order to justify the decisions made regarding the architecture of the model, some ablation studies have been carried out, which are presented below. These studies analyze the influence of the input data resolution (i.e. the number of vertices of the point clouds at the input of the model), the loss function used and some variations in the architecture of the model.

5.4.1 Input resolution

This first ablation study analyzes the influence of the point clouds resolutions on the model performance to predict an accurate saliency map. To do this, the proposed model, detailed in Section 4, was trained with the default parameters of the PointNet++ network and using four different resolutions: 20,000 vertices, 10,000 vertices, 5,000 vertices and 2,000 vertices.

Table 5.1 shows the numerical results obtained for the metrics explained above. As can

be seen, in general the range of values for the errors is very small. This happens because, as mentioned before, most of the values of the ground truth fixation maps are 0 or values very close to 0, so the model understands that the best way of reducing the error is by predicting very low values. The individual errors of each point cloud are shown in Appendix G.1.

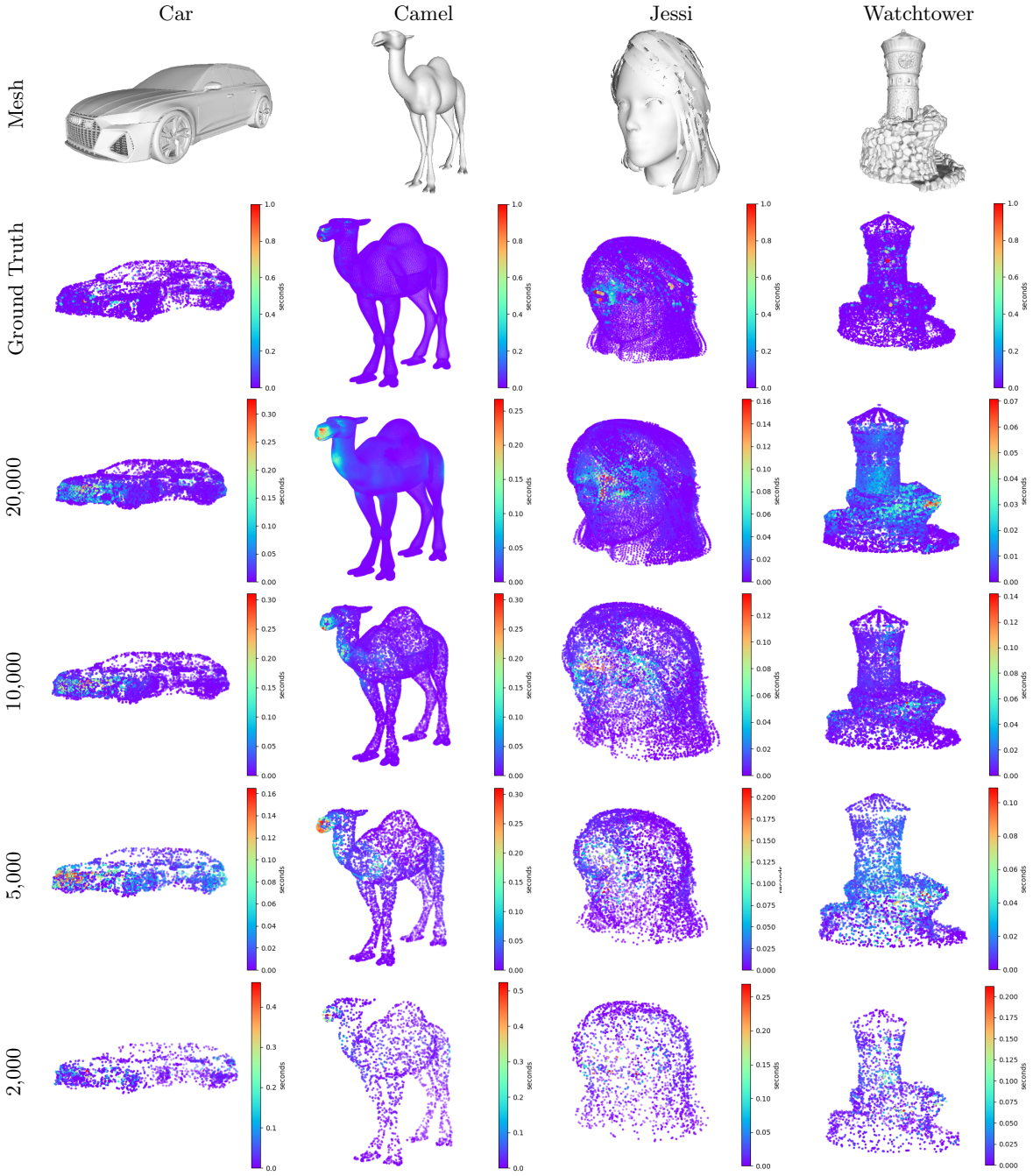


Figure 5.5: Comparison between the saliency of the ground truth and the predictions using different resolutions seen from the front view.

Furthermore, in Fig. 5.5 and Fig. 5.6 it can be seen how even though the computed errors shown in Table 5.1 for the different resolutions are different, the model is always consistent with

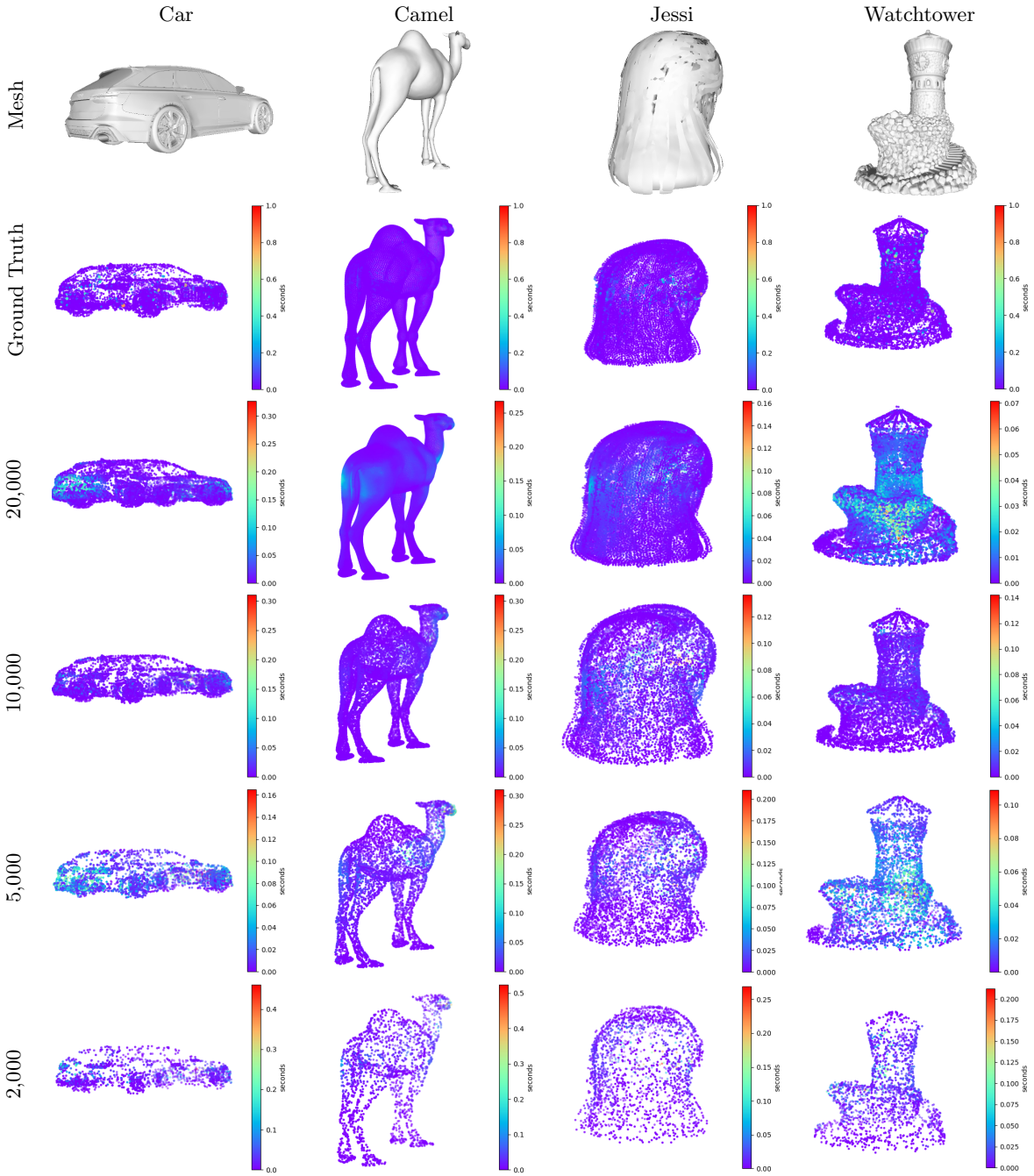


Figure 5.6: Comparison between the saliency of the ground truth and the predictions using different resolutions seen from the back view.

its predictions. It always predicts the same parts of the point clouds as interesting regardless of the number of vertices they have. It can also be confirmed that for those predictions where the model predicts a greater number of zeros (10,000 vertices and 2,000 vertices, especially on the watchtower), the error is smaller than for those predictions where there are more non-zero values (20,000 vertices and 5,000 vertices), which verifies that the model understands that for more accurate predictions, lower values are better.

Resolution	fixmaps = 0	MSE$\times 10^{-2}\downarrow$	MAE$\times 10^{-2}\downarrow$	CC$\uparrow$
20,000	w/	0.980	5.345	**0.364**
		(0.667)	(2.468)	**(0.198)**
	w/o	**1.335**	6.773	0.336
		(0.939)	(3.264)	(0.235)
10,000	w/	**0.508**	**3.238**	0.287
		(0.273)	**(0.744)**	(0.112)
	w/o	1.728	**6.522**	0.333
		(1.317)	**(2.612)**	(0.086)
5,000	w/	2.430	8.934	0.222
		(1.797)	(5.092)	(0.078)
	w/o	4.058	12.225	0.258
		(2.853)	(6.001)	(0.066)
2,000	w/	0.787	4.121	0.322
		(0.334)	(1.308)	(0.147)
	w/o	3.027	9.875	**0.349**
		(1.857)	(4.299)	**(0.176)**

Table 5.1: Comparison of metrics for the different point cloud resolutions. Arrows indicate whether higher or lower is better, and bold values highlight the best result for each metric, both considering all fixation maps (w/) and only the non-zero fixation maps (w/o). The values represent the mean score among the different test point clouds for each metric, and in brackets the averaged standard deviations are shown.

Respect to the Pearson's Correlation Coefficient, there is not a big relationship between the ground truth fixation maps and the predicted ones, but still, a value around 0.3 in 3D saliency prediction, and taking into account the large number of vertices the point clouds have, is a quite reasonable value. It has been then shown that the model is consistent regardless of the different resolutions, there is not a clear winner, the resuls of the errors and the CC depend on both the resolutions and the type of error to compute, as well as on if taking into account the fixation maps that are 0 or not. The final resolution selected has been 20,000 for having the largest CC.

Finally, the fact that the model is able to detect the interesting parts of the point clouds regardless of the number of vertices they have, together with the fact that there is no better resolution, corroborates what the authors of PointNet++ said about the density-adaptive PointNet layers they added in the *set abstraction* levels (see Section 4.1.1).

5.4.2 Alternative loss functions

For this metric and the next one (Section 5.4.3), since the metrics have been computed using the point clouds with all their vertices (i.e. 20,000), a smoothing based on the k nearest neighbours of each vertex has been applied to the point clouds in order to better approximate the appearance of human vision (see the details of the smoothing applied in Appendix F).

The results obtained for the metrics respect to the loss functions show that, again, the model is consistent with its preditions. The results are shown in Table 5.2, and it can be seen that there is hardly any difference between the errors. However, despite the Huber loss function gets slightly lower errors, the MSE loss function gets a better correlation, although again with a very

small difference. The individual errors of each point cloud are shown in Appendix G.2.

Fig. 5.7 and Fig. 5.8 show the results of the predictions using the different loss functions along with the smoothed ground truth. From the figures can be deduced that both the MSE and the Huber loss functions perform reasonably well. Nevertheless, the MAE loss function has not worked for our problem, the model did not learn anything during the training and the predictions for the test point clouds are all 0 (see the figures). That is the reason why the metrics are highlighted in red and there are no values in the Pearson's Correlation Coefficient. This happened because the Mean Absolute Error is robust to outliers, and in our case, it is precisely the different points that contain important information. Taking this into account, the final selected loss function has been the MSE for having the largest CC.

Loss function	fixmaps = 0	MSE$\times 10^{-2}\downarrow$	MAE$\times 10^{-2}\downarrow$	CC$\uparrow$
MSELoss()	w/	0.980	5.345	**0.364**
		(0.667)	(2.468)	**(0.198)**
	w/o	1.335	6.773	**0.336**
		(0.939)	(3.264)	**(0.235)**
MAELoss()	w/	0.553	3.271	-
		(0.150)	(0.358)	-
	w/o	0.986	5.525	-
		(0.554)	(2.021)	-
HuberLoss()	w/	**0.843**	**4.953**	0.342
		(0.560)	**(1.957)**	(0.170)
	w/o	**1.228**	**5.964**	0.319
		(1.116)	**(3.000)**	(0.210)

Table 5.2: Comparison of metrics using different loss functions for training. Arrows indicate whether higher or lower is better, and bold values highlight the best result for each metric, both considering all fixation maps (w/) and only the non-zero fixation maps (w/o). The values represent the mean score among the different test point clouds for each metric, and in brackets the averaged standard deviations are shown.

5.4.3 Model architecture variations

Finally, the last ablation study consisted of changing some architecture parameters to the network and see how this affected the performance of the model. PointNet++ allows to modify some architecture parameters for the training such as the number of local regions K, the number of nearest neighbours between which to interpolate features knn or the radius of the ball in the Ball Query algorithm r. For this project, the parameters changed in order to see the performance of the model were K and knn, whose default values were 64 and 3, respectively.

The numerical results obtained for the metrics with the different architecture configurations are presented in Table 5.3. As can be seen, and as mentioned in the previous ablation studies, the errors are again in a very low range of values and are not very different from each other. There is no clear winner either, since depending on the metric and whether the ground truth fixation maps with value 0 are taken into account or not, the best architecture varies. The individual errors of each point cloud are shown in Appendix G.3.

In addition, to show that there is not a clear winner, Fig. 5.9 and Fig. 5.10 show the predic-

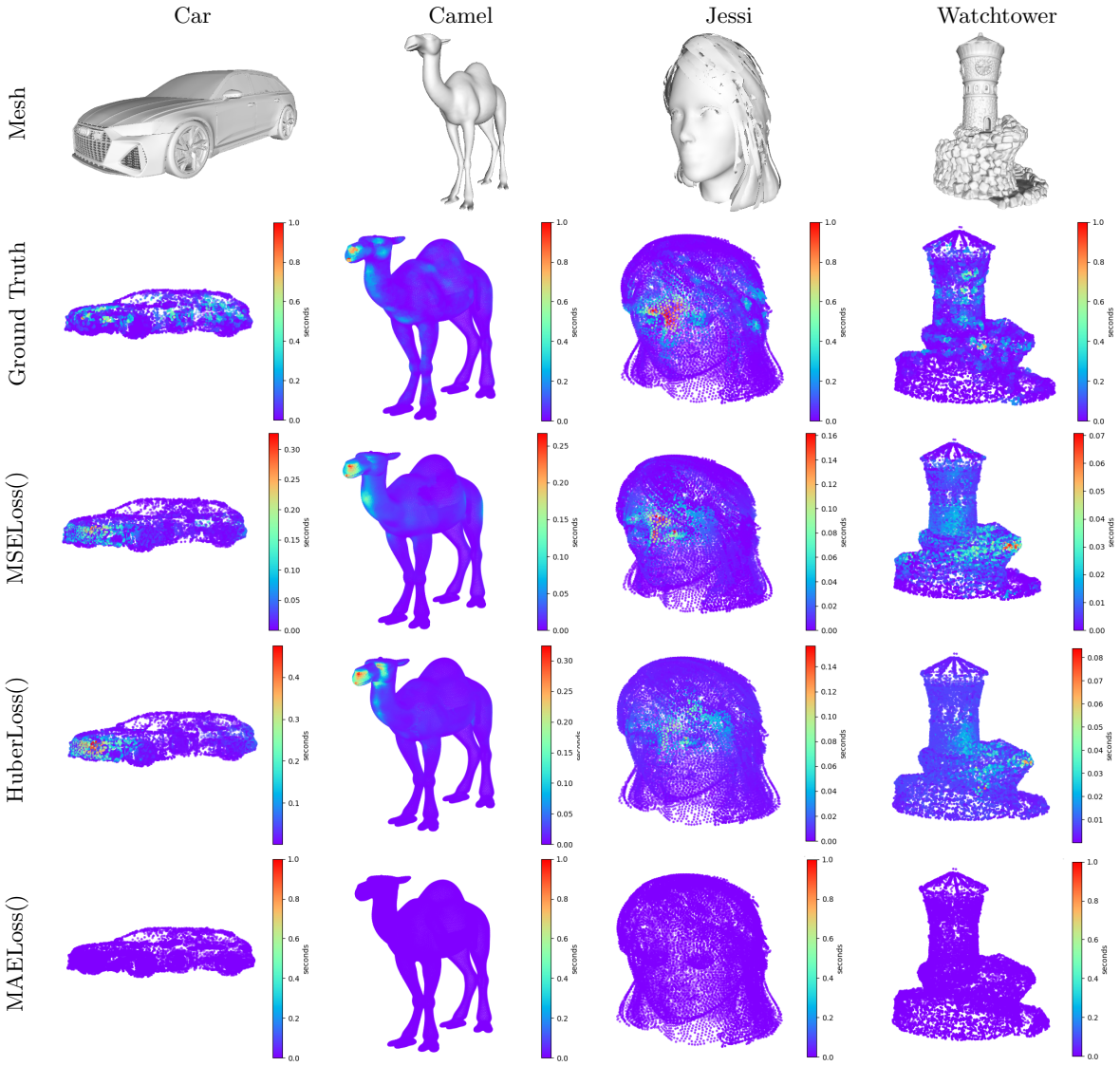


Figure 5.7: Comparison between the saliency of the ground truth and the predictions using different loss functions seen from the front view.

tions on the test point clouds with the different network architecture configurations. It can be seen how the network is again able to recognize and detect the most interesting parts of each point cloud. Depending on each point cloud, there are some configurations that predict these parts better than others. For example, according to the quantitative results of the errors shown in Table 5.3, the best architecture is the one with $K = 64$ and $knn = 50$, since most of the values of the ground truth fixation maps are 0 and this configuration predicts a more quantity of zeros on the test point clouds than the other configurations, so errors are smaller. However, qualitatively, looking at Fig. 5.9 and Fig. 5.10, this reasoning is valid for the point clouds of the car, the camel and the girl, but not for the watchtower. The predictions for the watchtower with this configuration are all zeros except one only point, which does not match the ground truth. So, for this mesh, the configuration that best matches the ground truth is the one with $K = 512$ and $knn = 3$. Taking all this into account, the final parameters selected have been $K = 64$ and

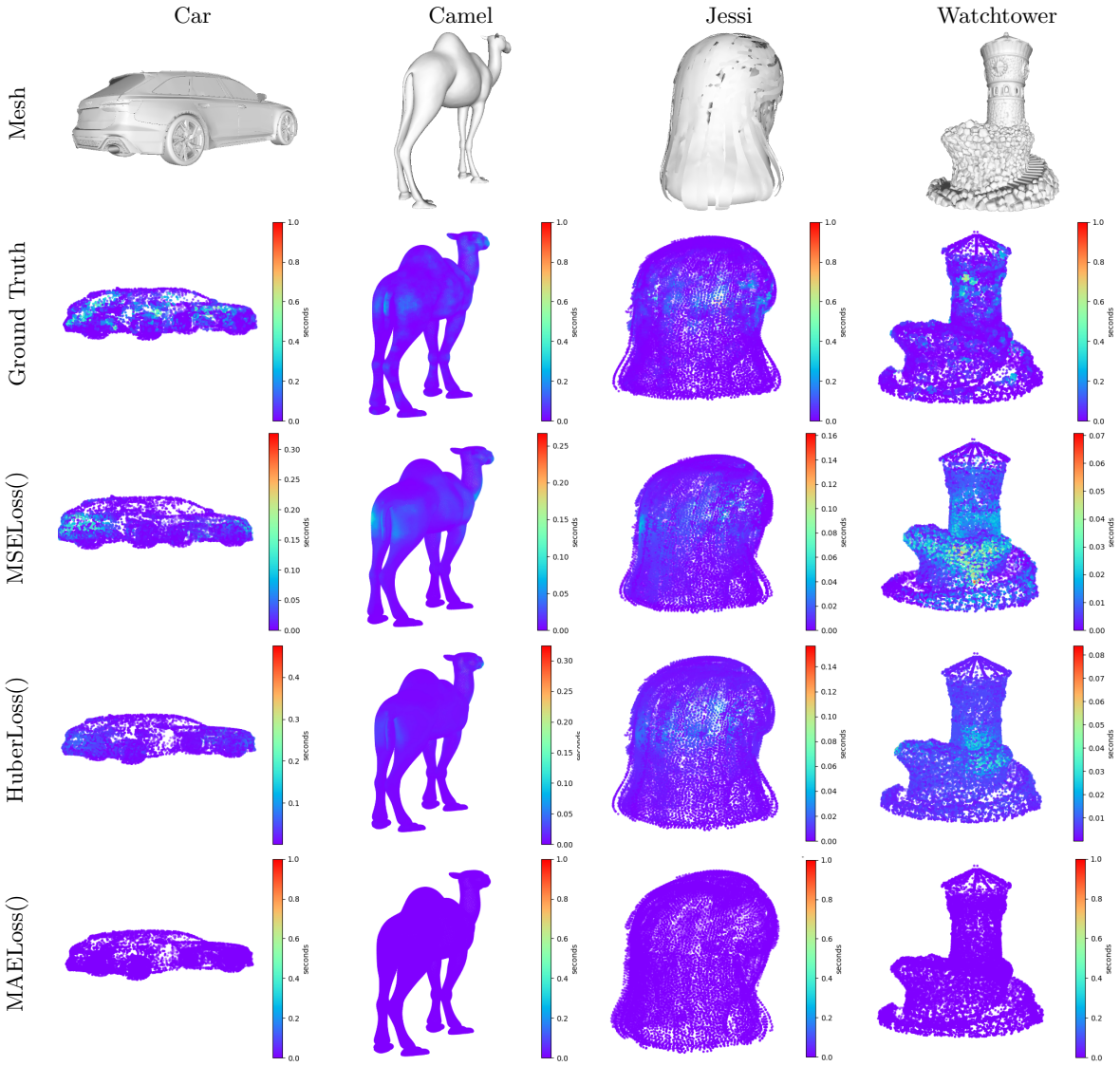


Figure 5.8: Comparison between the saliency of the ground truth and the predictions using different loss functions seen from the back view.

$knn = 3$ for having the largest CC.

K	knn	fixmaps = 0	MSE$\times 10^{-2}\downarrow$	MAE$\times 10^{-2}\downarrow$	CC$\uparrow$
64	3	w/	0.980	5.345	**0.364**
			(0.667)	(2.468)	**(0.198)**
		w/o	1.335	6.773	**0.336**
			(0.939)	(3.264)	**(0.235)**
64	50	w/	**0.574**	**3.382**	0.265
			(0.302)	**(0.623)**	(0.194)
		w/o	**1.028**	5.656	0.251
			(0.789)	(2.495)	(0.208)
512	3	w/	0.646	3.603	0.319
			(0.312)	(0.983)	(0.181)
		w/o	1.056	**5.644**	0.291
			(0.718)	**(2.535)**	(0.215)
512	50	w/	0.801	3.949	0.361
			(0.521)	(1.105)	(0.166)
		w/o	1.318	6.011	0.326
			(1.130)	(2.868)	(0.202)

Table 5.3: Comparison of metrics for the different network architecture configurations. Arrows indicate whether higher or lower is better, and bold values highlight the best result for each metric, both considering all fixation maps (w/) and only the non-zero fixation maps (w/o). The values represent the mean score among the different test point clouds for each metric, and in brackets the averaged standard deviations are shown.

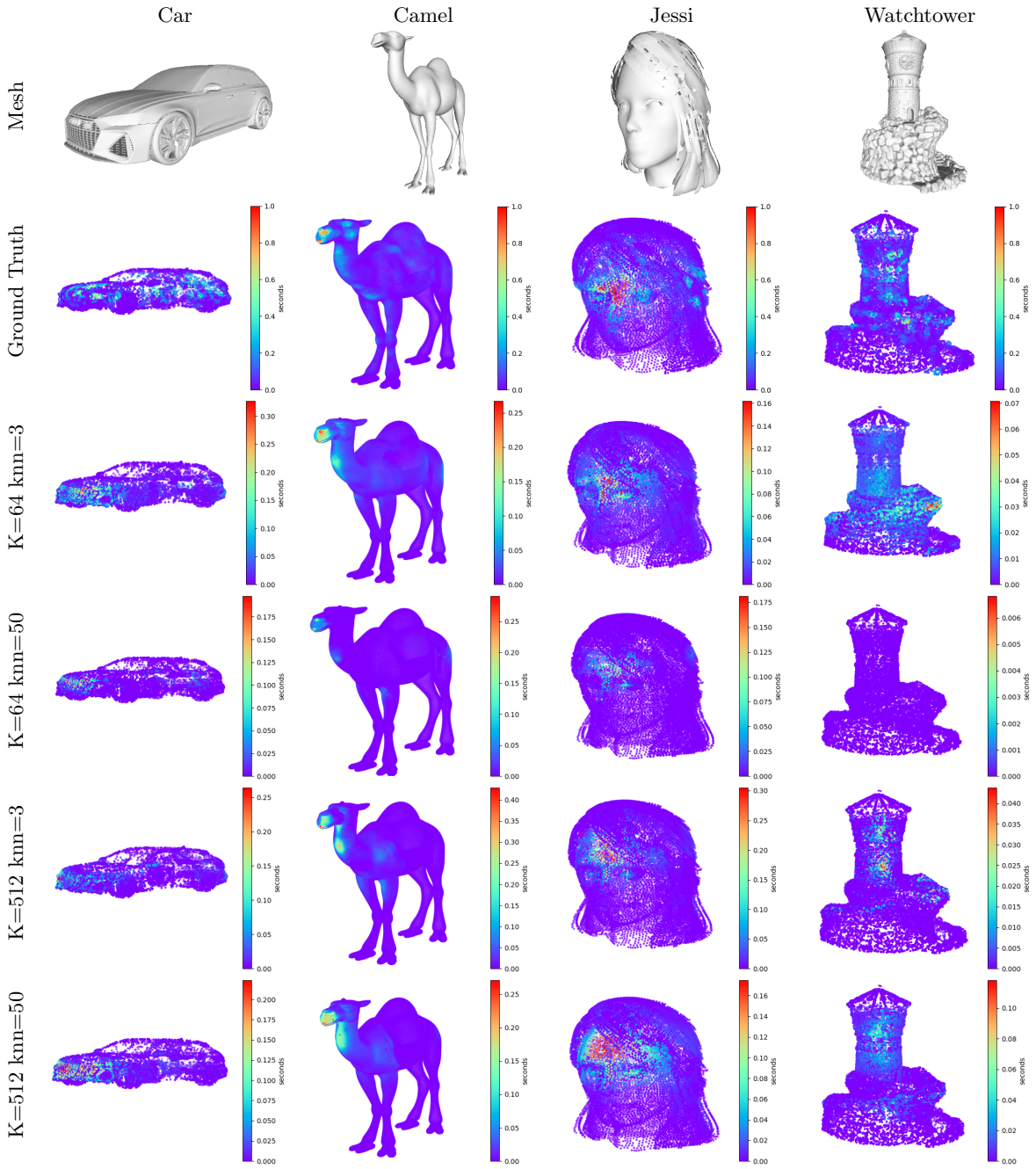


Figure 5.9: Comparison between the saliency of the ground truth and the predictions using different networks architectures seen from the front view.

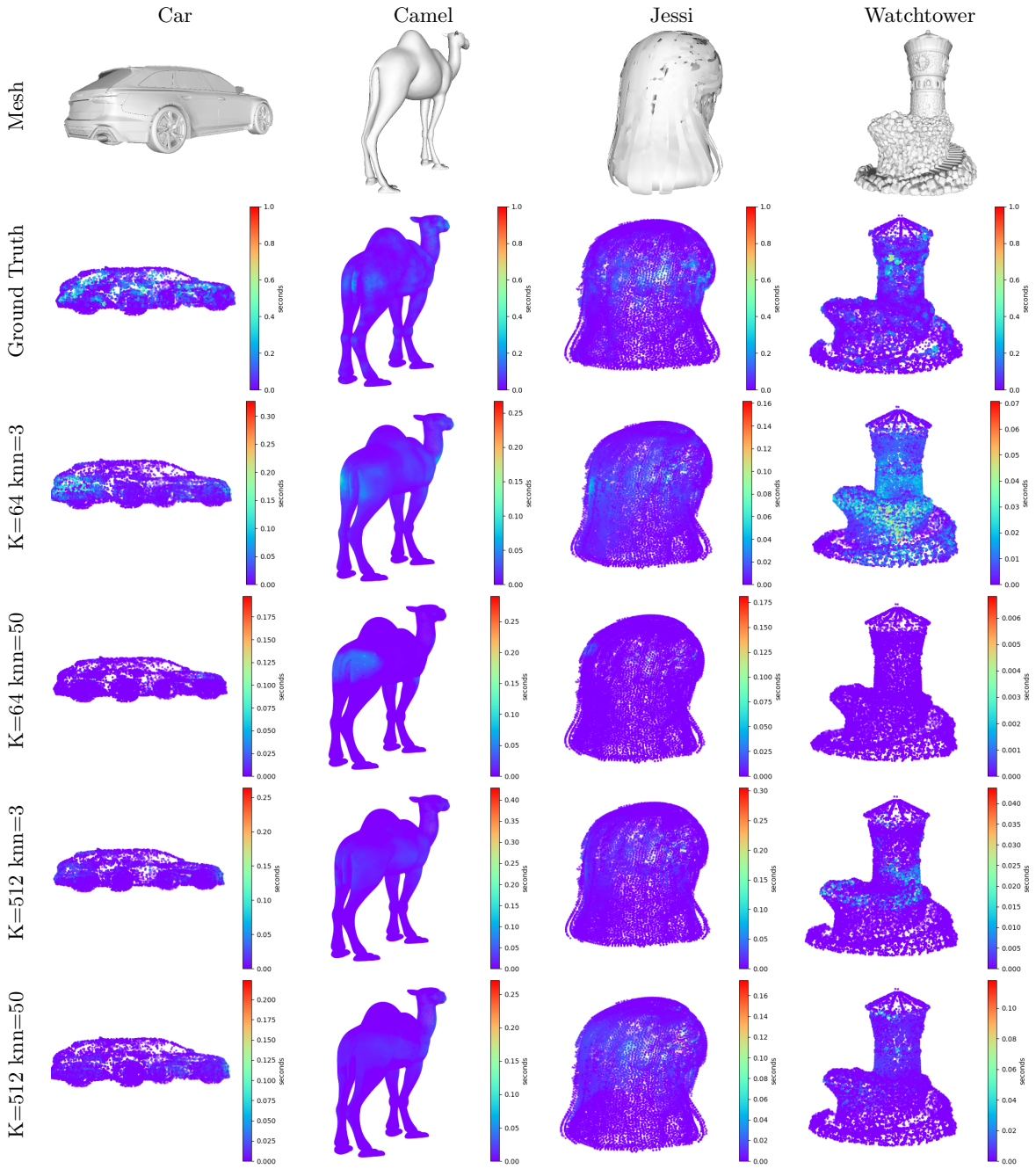


Figure 5.10: Comparison between the saliency of the ground truth and the predictions using different networks architectures seen from the back view.

6. Conclusions

This Master's thesis has presented a deep learning-based approach to model human visual attention behavior when viewing 3D objects through saliency prediction. The proposed saliency prediction model has been adapted to point clouds, taking into account information from local and global regions of these point clouds. Also, it has been demonstrated that our approach has been able to generalize the areas of interest of 3D objects reasonably well and obtains accurate saliency predictions in a wide variety of objects.

The fact that no works have been found that try to predict saliency using real gaze data as ground truth at the time of doing this thesis proves that this is a challenging task. Because of this, we decided to carry out this project. At first, we started to work with the available saliency datasets provided by Lavoué et al. [13] and Wang et al. [14], but their sizes were not large enough to obtain reasonable results, so an experiment was carried out in order to create our own dataset, with a size larger than the size of the existing ones. The results obtained showed that the decision was correct, since increasing the dataset, the model improved the saliency prediction task and it was possible to compare our results with others from other works.

In addition, based on the results obtained and the ablation study carried out, it can be deduced that, with our dataset, the best possible results were accomplished, since there is not a big difference between the results obtained neither for the different point cloud resolutions, nor when using the different loss functions nor changing the network architecture parameters.

Finally, according to the comparison of our work with other works, it has been shown that our approach is capable of obtaining saliency prediction results that are close to the obtained ones by other approaches, which is a great achievement in the field since our work has been carried out with real gaze data as ground truth without making any assumption regarding the feature vectors extracted by the network in its hidden layers or the steep curvatures and corners of the meshes.

6.1 Limitations and future work

The work carried out in this master's thesis aims to be a first step towards understanding and modeling human visual attention behavior when viewing 3D objects using deep learning and real gaze data as ground truth, with the goal of serving as a basis for future work in this field, which remains an open problem. There is room for improvement in point cloud saliency prediction and further work is expected.

Regarding the saliency prediction results obtained, and taking into account the fact that increasing the dataset improved the performance of our model, this dataset could be extended in future work to see if the performance continues to improve. In general, all deep learning models tend to perform better the more data they have to learn, since the size of the dataset is usually the main limitation for all deep learning models. So, intuitively, extending the dataset further should improve the performance of the model.

Another limitation of this project is that the model was trained with the raw fixations maps instead of with saliency maps, i.e. our ground truth are the fixation maps extracted from the experiment aggregated and normalized between 0 and 1, so there are only a few vertices that have a fixation and it is very difficult to exactly match the predictions. For example, looking at the second row of Fig. 5.9 it can be seen that in the ground truth there are only a few points fixated, so this could be limiting the performance of the model. However, creating a saliency map in 3D, and more specifically in a point cloud, is not an easy task, since to do it correctly it is necessary to take into account the user position, the gaze direction, the distance to the collision point, etc, and then it is necessary to replace the gaze ray by a cone and to project a Gaussian distribution on the point cloud. So, as future work, creating the saliency maps and training the model with them instead of with the raw fixation maps might also improve the performance of the model.

Another possible extension for this work could also be saliency prediction but on virtual scenes. That is, designing virtual scenes (e.g. a living room) and allowing participants to freely move and inspect the room. The gaze data would be collected and then a deep learning-based model could be trained to predict interesting parts of a scene instead of an only object.

Finally, another aspect to take into account could be the previous knowledge of the participants. Human visual attention behaviour is often influenced by prior knowledge. Therefore, how to integrate these prior knowledge into the neural network could be useful to obtain more accurate results.

Bibliography

[1] Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 5105–5114, Red Hook, NY, USA, 2017. Curran Associates Inc.

[2] Ran Song, Yonghuai Liu, and Paul L. Rosin. Mesh saliency via weakly supervised classification-for-saliency cnn. *IEEE Transactions on Visualization and Computer Graphics*, 27(1):151–164, 2021.

[3] Chengming Liu, Wan-na Luan, Rong-hua Fu, Hai-bo Pang, and Ying-hao Li. Attention-embedding mesh saliency. *The Visual Computer*, may 2022.

[4] Jinho Lee, B. Moghaddam, H. Pfister, and R. Machiraju. Finding optimal views for 3d face shape modeling. In *Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. Proceedings.*, pages 31–36, 2004.

[5] Ran Gal and Daniel Cohen-Or. Salient geometric features for partial shape matching and similarity. *ACM Trans. Graph.*, 25(1):130–150, jan 2006.

[6] Philip Shilane and Thomas Funkhouser. Distinctive regions of 3d surfaces. *ACM Trans. Graph.*, 26(2):7–es, jun 2007.

[7] Yu-Bin Yang, Tong Lu, and Jin-Jie Lin. Saliency regions for 3d mesh abstraction. In *Proceedings of the 10th Pacific Rim Conference on Multimedia: Advances in Multimedia Information Processing*, PCM '09, page 292–299, Berlin, Heidelberg, 2009. Springer-Verlag.

[8] Chang Ha Lee, Amitabh Varshney, and David W. Jacobs. Mesh saliency. *ACM Trans. Graph.*, 24(3):659–666, jul 2005.

[9] George Leifman, Elizabeth Shtrom, and Ayellet Tal. Surface regions of interest for viewpoint selection. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 414–421, 2012.

[10] Ran Song, Yonghuai Liu, Ralph R. Martin, and Paul L. Rosin. Mesh saliency via spectral processing. *ACM Trans. Graph.*, 33(1), feb 2014.

[11] Flora Ponjou Tasse, Jiri Kosinka, and Neil Dodgson. Cluster-based point set saliency. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 163–171, 2015.

[12] Stavros Nousias, Gerasimos Arvanitis, Aris S. Lalos, and Konstantinos Moustakas. Mesh saliency detection using convolutional neural networks. In *2020 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2020.

[13] Guillaume Lavoué, Frédéric Cordier, Hyewon Seo, and Mohamed-Chaker Larabi. Visual attention for rendered 3d shapes. *Computer Graphics Forum*, 37(2):191–203, 2018.

[14] Xi Wang, Sebastian Koch, Kenneth Holmqvist, and Marc Alexa. Tracking the gaze on objects in 3d: How do people really look at the bunny? *ACM Trans. Graph.*, 37(6), dec 2018.

[15] Xiaobai Chen, Abulhair Saparov, Bill Pang, and Thomas Funkhouser. Schelling points on 3d surface meshes. *ACM Trans. Graph.*, 31(4), jul 2012.

[16] Vincent Sitzmann, Semon Rezchikov, William T. Freeman, Joshua B. Tenenbaum, and Fredo Durand. Light field networks: Neural scene representations with single-evaluation rendering. In *Proc. NeurIPS*, 2021.

[17] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

[18] Amit Kohli, Vincent Sitzmann, and Gordon Wetzstein. Semantic implicit neural scene representations with semi-supervised training. *2020 International Conference on 3D Vision (3DV)*, pages 423–433, 2020.

[19] Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *arXiv*, 2020.

[20] Bing Liu, Weihua Dong, and Liqiu Meng. Using eye tracking to explore the guidance and constancy of visual variables in 3d visualization. *ISPRS International Journal of Geo-Information*, 6(9), 2017.

[21] Sangwon Lee, Eungee Cinn, Jin Yan, and Jaeyoon Jung. Using an eye tracker to study three-dimensional environmental aesthetics: The impact of architectural elements and educational training on viewers' visual attention. *Journal of architectural and planning research*, 32:145–167, 06 2015.

[22] Sarah Howlett, John Hamill, and Carol O'Sullivan. Predicting and evaluating saliency for simplified polygonal models. *ACM Trans. Appl. Percept.*, 2(3):286–308, jul 2005.

[23] Xi Wang, David Lindlbauer, Christian Lessig, and Marc Alexa. Accuracy of monocular gaze tracking on 3d geometry. In *Eye Tracking and Visualization*, pages 169–184, 02 2017.

[24] Xi Wang, David Lindlbauer, Christian Lessig, Marianne Maertens, and Marc Alexa. Measuring the visual salience of 3d printed objects. *IEEE Computer Graphics and Applications*, 36(4):46–55, 2016.

[25] Tilke Judd, Krista Ehinger, Frédo Durand, and Antonio Torralba. Learning to predict where humans look. In *2009 IEEE 12th International Conference on Computer Vision*, pages 2106–2113, 2009.

[26] Youngmin Kim, Amitabh Varshney, David W. Jacobs, and François Guimbretière. Mesh saliency and human eye fixations. *ACM Trans. Appl. Percept.*, 7(2), feb 2010.

[27] Vincent Sitzmann, Ana Serrano, Amy Pavel, Maneesh Agrawala, Diego Gutierrez, Belen Masia, and Gordon Wetzstein. Saliency in VR: How do people explore virtual environments? *IEEE Transactions on Visualization and Computer Graphics*, 24(4):1633–1642, 2018.

[28] Evangelos Alexiou, Peisen Xu, and Touradj Ebrahimi. Towards modelling of visual saliency in point clouds for immersive applications. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 4325–4329, 2019.

[29] Ana Serrano, Vincent Sitzmann, Jaime Ruiz-Borau, Gordon Wetzstein, Diego Gutierrez, and Belen Masia. Movie editing and cognitive event segmentation in virtual reality video. *ACM Transactions on Graphics (SIGGRAPH 2017)*, 36(4), 2017.

[30] Christof Koch and Shimon Ullman. Shifts in selective visual attention: towards the underlying neural circuitry. *Human neurobiology*, 4(4):219–27, 1985.

[31] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, 1998.

[32] Marcella Cornia, Lorenzo Baraldi, Giuseppe Serra, and Rita Cucchiara. A Deep Multi-Level Network for Saliency Prediction. In *International Conference on Pattern Recognition (ICPR)*, 2016.

[33] Xun Huang, Chengyao Shen, Xavier Boix, and Qi Zhao. Salicon: Reducing the semantic gap in saliency prediction by adapting deep neural networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 262–270, 2015.

[34] Guanbin Li and Yizhou Yu. Visual saliency based on multiscale deep features. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5455–5463, 06 2015.

[35] Junting Pan, Elisa Sayrol, Xavier Giro-i Nieto, Kevin McGuinness, and Noel E. O'Connor. Shallow and deep convolutional networks for saliency prediction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[36] Lijun Wang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Deep networks for saliency detection via local estimation and global search. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3183–3192, 2015.

[37] Rui Zhao, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Saliency detection by multi-context deep learning. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1265–1274, 2015.

[38] Lai Jiang, Zhe Wang, Mai Xu, and Zulin Wang. Image saliency prediction in transformed domain: A deep complex neural network method. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):8521–8528, Jul. 2019.

[39] Blanca Lasheras-Hernandez, Belen Masia, and Daniel Martin. Drivernn: Predicting drivers' attention with deep recurrent networks. In *Spanish Computer Graphics Conference (CEIG)*, 2022.

[40] Souad Chaabouni, Jenny Benois-Pineau, Ofer Hadar, and Chokri Ben Amar. Deep learning for saliency prediction in natural video, 2016.

[41] Marc Assens, Xavier Giro-i Nieto, Kevin McGuinness, and Noel E. O'Connor. Saltinet: Scan-path prediction on 360 degree images using saliency volumes. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 2331–2338, 2017.

[42] Daniel Martin, Ana Serrano, Alexander W. Bergman, Gordon Wetzstein, and Belen Masia. Scangan360: A generative model of realistic scanpaths for 360º images. *IEEE Transactions on Visualization and Computer Graphics*, 28(5):2003–2013, 2022.

[43] Daniel Martin, Ana Serrano, and Belen Masia. Panoramic convolutions for 360° single-image saliency prediction. In *CVPR Workshop on Computer Vision for Augmented and Virtual Reality*, 2020.

[44] Manfred Lau, Kapil Dev, Weiqi Shi, Julie Dorsey, and Holly Rushmeier. Tactile mesh saliency. *ACM Trans. Graph.*, 35(4), jul 2016.

[45] Jinliang Wu, Xiaoyong Shen, Wei Zhu, and Ligang Liu. Mesh saliency with global rarity. *Graph. Models*, 75(5):255–264, sep 2013.

[46] Shanfeng Hu, Xiaohui Liang, Hubert P. H. Shum, Frederick W. B. Li, and Nauman Aslam. Sparse metric-based mesh saliency. *Neurocomputing*, 400:11–23, 2020.

[47] M. Limper, A. Kuijper, and D. W. Fellner. Mesh saliency analysis via local curvature entropy. In *Proceedings of the 37th Annual Conference of the European Association for Computer Graphics: Short Papers*, EG '16, page 13–16, Goslar, DEU, 2016. Eurographics Association.

[48] Ning Wei, Kaiyuan Gao, Rongrong Ji, and Peng Chen. Surface saliency detection based on curvature co-occurrence histograms. *IEEE Access*, 6:54536–54541, 2018.

[49] Umberto Castellani, Marco Cristani, Simone Fantoni, and Vittorio Murino. Sparse points matching by combining 3d mesh saliency with statistical descriptors. *Comput. Graph. Forum*, 27:643–652, 04 2008.

[50] Xiaoying Ding, Weisi Lin, Zhenzhong Chen, and Xinfeng Zhang. Point cloud saliency detection by local and global feature fusion. *IEEE Transactions on Image Processing*, 28(11):5379–5393, 2019.

[51] Tianhang Zheng, Changyou Chen, Junsong Yuan, Bo Li, and Kui Ren. Pointcloud saliency maps. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1598–1606, 2019.

[52] Jing Huang and Suya You. Point cloud labeling using 3d convolutional neural network. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 2670–2675, 2016.

[53] R. Qi Charles, Hao Su, Mo Kaichun, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 77–85, 2017.

[54] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[55] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph cnn for learning on point clouds. *ACM Trans. Graph.*, 38(5), oct 2019.

[56] Zhirong Wu, Shuran Song, et al. 3d shapenets: A deep representation for volumetric shapes. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1912–1920, 2015.

[57] Angel X. Chang et al. Shapenet: An information-rich 3d model repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University, Princeton University, Toyota Technological Institute at Chicago, 2015.

[58] Iro Armeni, Sasha Sax, Amir R Zamir, and Silvio Savarese. Joint 2d-3d-semantic data for indoor scene understanding. *arXiv preprint arXiv:1702.01105*, 2017.

[59] Angela Dai et al. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

[60] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

[61] Zeke Xie, Issei Sato, and Masashi Sugiyama. Understanding and scheduling weight decay, 2020.

[62] Pingping Tao, Junjie Cao, Shuhua Li, Xiuping Liu, and Ligang Liu. Mesh saliency via ranking unsalient patches in a descriptor space. *Comput. Graph.*, 46(C):264–274, feb 2015.

Appendix A. Informed consent

At the beginning of the experiment, participants fill and sign the informed consent, required by the government of Aragón, which states the aim of the experiment as well as the rights of the participants. A copy of this document is attached on the next page.

DOCUMENTO DE INFORMACIÓN PARA EL PARTICIPANTE

Título de la investigación: Modelos de aprendizaje profundo para predicción de saliencia en objetos 3D

Promotor: Universidad de Zaragoza

Investigador Principal: xxxxxxxxxxxxxxxxxxx **Tfno:** XXXXXXXXX **mail: xxxxxxxxxx@xxxxxxxxxx**

Centro: Edificio I3A, C/Mariano Esquillor s/n, campus Río Ebro, 50018, Zaragoza

1. Introducción:

Nos dirigimos a usted para solicitar su participación en un proyecto de investigación que estamos realizando en la Universidad de Zaragoza. Su participación es absolutamente voluntaria, en ningún caso debe sentirse obligado a participar, pero es importante para obtener el conocimiento que necesitamos. Este proyecto ha sido aprobado por el Comité de Ética de la Investigación de la Comunidad Autónoma de Aragón. Antes de tomar una decisión es necesario que:

- lea este documento entero
- entienda la información que contiene el documento
- haga todas las preguntas que considere necesarias
- tome una decisión meditada
- firme el consentimiento informado, si finalmente desea participar.

Si decide participar se le entregará una copia de esta hoja y del documento de consentimiento firmado. Por favor, consérvelo por si lo necesitara en un futuro.

2. ¿Por qué se le pide participar?

Se le solicita su colaboración porque tiene visión normal o corregida y se ha prestado voluntario libremente para realizar un experimento en el cual se estudiará hacia donde se dirige su atención visual en una animación generada por ordenador. Para la realización de este estudio no se requiere ningún tipo de conocimiento previo del campo.
En total, en el estudio participarán 32 personas.

3. ¿Cuál es el objeto de este estudio?

El objetivo de este estudio es el diseño y entrenamiento de un modelo de aprendizaje profundo capaz de predecir hacia donde dirigiría, mayoritariamente, la atención una persona al visualizar un objeto 3D. Para ello será necesario recopilar datos de atención visual humana en diversos objetos 3D. Estos datos se van a recopilar utilizando unas gafas de realidad virtual HTC Vive, el usuario se las pondrá y en el entorno de realidad virtual diseñado para la realización del experimento varios objetos, uno detrás de otro, aparecerán en frente del participante, harán una rotación completa sobre sí mismos, y se recopilarán los datos de a qué puntos de los objetos el participante ha dirigido su mirada. El resultado esperado es el diseño de un modelo de aprendizaje profundo capaz de predecir hacia donde dirigiría la atención un humano en un objeto en la primera vista de este.

4. ¿Qué tengo que hacer si decido participar?

Si decide participar, deberá observar un total de 60 objetos 3D que aparecerán en frente en el entorno de realidad virtual diseñado para la realización del experimento, uno detrás de otro, haciendo una rotación completa sobre sí mismos, utilizando para ello las gafas de realidad virtual HTC Vive. Así mismo, deberá intentar mover la cabeza lo menos posible. El experimento se dividirá en dos sesiones, cada una de unos 30 minutos de duración aproximadamente, y deberá rellenar dos formularios:

- Uno demográfico antes de empezar experimento, en el que se le preguntará por la edad, el sexo, sus patologías visuales, si tiene visión normal o corregida con gafas o lentillas, si juega a videojuegos habitualmente y si ha utilizado alguna vez algún dispositivo de realidad virtual.
- Uno al finalizar el experimento con las sensaciones, en el que se le pedirá que puntúe el experimento del 1 al 5 en función de cuanto le haya gustado (siendo el 1 nada y el 5 mucho), se le preguntará si durante el experimento ha tenido la sensación de estar presente en el entorno virtual creado para el experimento y hasta qué punto ha sido consciente de los ruidos y distracciones del mundo real.

Esta información nos servirá para analizar si una peor calidad en el resultado del experimento para un participante puede atribuirse a edad avanzada o a problemas de visión (por ejemplo, mala visión de cerca o estrabismo) y para saber la opinión personal de los participantes acerca del experimento.

5. ¿Qué riesgos o molestias supone?

Debido a la naturaleza del experimento, no se espera ningún riesgo o molestia significativa por parte de los participantes. Aun así, debido a que el experimento se va a realizar utilizando un dispositivo de realidad virtual, el participante podría experimentar un leve, aunque improbable, mareo que cesa en pocos minutos, ya que se trata de una de las principales consecuencias del uso de dispositivos de realidad virtual.

6. ¿Obtendré algún beneficio por mi participación?

Al tratarse de un estudio de investigación orientado a generar conocimiento no es probable que obtenga ningún beneficio por su participación si bien usted contribuirá al avance científico y al beneficio social.

7. ¿Cómo se van a tratar mis datos personales?

Información básica sobre protección de datos.

Responsable del tratamiento: Andrés Fandos Villanueva

Finalidad: Sus datos personales serán tratados exclusivamente para el trabajo de investigación a los que hace referencia este documento.

Legitimación: El tratamiento de los datos de este estudio queda legitimado por su consentimiento a participar.

Destinatarios: No se cederán datos a terceros salvo obligación legal.

Duración: La recogida de datos en la que solicitamos su colaboración forma parte del Trabajo Fin de Máster del estudiante Andrés Fandos Villanueva. Los datos serán destruidos a la finalización del Trabajo Fin de Máster, el 31 de diciembre de 2022, ya que esta es la fecha límite para la defensa del proyecto. Los resultados objeto de explotación, ya completamente anonimizados y sin datos personales, podrán ser conservados para su posible reutilización en otros trabajos de investigación.

Derechos: Podrá ejercer sus derechos de acceso, rectificación, supresión y portabilidad de sus datos, de limitación y oposición a su tratamiento, de conformidad con lo dispuesto en la LO 3/2018 de Protección de Datos Personales y garantía de los derechos digitales y el Reglamento General de Protección de Datos (RGPD 2016/679) ante el investigador principal del proyecto, cuyos datos de contacto figuran en el encabezamiento de este documento.

Así mismo, en cumplimiento de los dispuesto en el RGPD, se informa que, si así lo desea, podrá acudir a la Agencia de Protección de Datos (https://www.aepd.es) para presentar una reclamación cuando considere que no se hayan atendido debidamente sus derechos. Podrá consultar también información adicional y detallada de este tratamiento de datos en el Inventario de Actividades de Tratamiento de la Universidad de Zaragoza, así como en el siguiente enlace: https://protecciondatos.unizar.es o enviando un email a la dirección de la unidad de protección de datos de la Universidad de Zaragoza: dpd@unizar.es.

El tratamiento de sus datos personales se realizará utilizando técnicas para mantener su anonimato mediante el uso de códigos aleatorios, con el fin de que su identidad personal quede completamente oculta durante el proceso de investigación.

A partir de los resultados del trabajo de investigación, se podrán elaborar comunicaciones científicas para ser presentadas en congresos o revistas científicas, pero se harán siempre con datos agrupados y nunca se divulgará nada que le pueda identificar.

9. ¿Quién financia el estudio?

Este proyecto no cuenta con financiación.

10. ¿Se me informará de los resultados del estudio?

Usted tiene derecho a conocer los resultados del presente estudio, tanto los resultados generales como los derivados de sus datos específicos. También tiene derecho a no conocer dichos resultados si así lo desea. Por este motivo en el documento de consentimiento informado le preguntaremos qué opción prefiere. En caso de que desee conocer los resultados, el investigador se los hará llegar.

¿Puedo cambiar de opinión?

Su participación es totalmente voluntaria, puede decidir no participar o retirarse del estudio en cualquier momento sin tener que dar explicaciones. Basta con que le manifieste su intención al investigador principal del estudio. En caso de que decida retirarse del estudio puede solicitar la destrucción de los datos, muestras u otra información recogida sobre usted.

¿Qué pasa si me surge alguna duda durante mi participación?

En la primera página de este documento está recogido el nombre y el teléfono de contacto del investigador responsable del estudio. Puede dirigirse a él en caso de que le surja cualquier duda sobre su participación.

Muchas gracias por su atención, si finalmente desea participar le rogamos que firme el documento de consentimiento que se adjunta y le reiteramos nuestro agradecimiento por contribuir a generar conocimiento científico.

DOCUMENTO DE CONSENTIMIENTO INFORMADO

Título del PROYECTO: Modelos de aprendizaje profundo para predicción de saliencia en objetos 3D

Yo, ... (nombre y apellidos del participante)

He leído la hoja de información que se me ha entregado.
He podido hacer preguntas sobre el estudio y he recibido suficiente información sobre el mismo.
He hablado con: …….. (nombre del investigador)
Comprendo que mi participación es voluntaria.
Comprendo que puedo retirarme del estudio:

1) cuando quiera
2) sin tener que dar explicaciones
3) sin que esto repercuta en mis cuidados médicos

Presto libremente mi consentimiento para participar en este estudio y doy mi consentimiento para el acceso y utilización de mis datos conforme se estipula en la hoja de información que se me ha entregado.

Deseo ser informado sobre los resultados del estudio: SI □ NO □ (marque lo que proceda)

Si marca SÍ indique su teléfono o correo electrónico de contacto: _____

He recibido una copia firmada de este Consentimiento Informado.

Firma del participante: .....
Fecha: .....

He explicado la naturaleza y el propósito del estudio al paciente mencionado

Firma del Investigador: .....
Fecha:

Appendix B. Demographic questionnaire

After filling out the informed consent (Appendix A) users fulfilled the following demographic questionnaire. Details of the data collected with this questionnaire are shown in Section 3.2. A copy of this questionnaire is attached on the next page.

Demographic questionnaire

Questionnaire to provide consent, collect some demographic information, and to ask participants if they had any experience with video-games, HMDs and VR environments.

*Obligatorio

1. Subject ID *

2. I agree to participate in this research study. I understand the purpose and nature of this study and am participating voluntarily. I understand that I may withdraw from the study at any time without penalty or consequence. I consent to the use of the data generated from this questionnaire in the researcher's publications on this topic. Any information obtained in connection with this study that can be identified with you will remain confidential and will be released only with your permission. (https://policies.google.com/privacy) *

   Selecciona todos los que correspondan.

   - [ ] I agree.

3. Age *

4. Gender *

Marca solo un óvalo.

- Male
- Female
- Non-Binary
- Prefer not to say
- Otro: _____

5. Do you have any vision problems? *

Marca solo un óvalo.

- Yes
- No

6. If yes, what type of problem is it (e.g. poordistance vision, astigmatism, etc.)?

_____

_____

_____

_____

_____

7. If yes, do you have them corrected wearing glasses or contact lenses?

Marca solo un óvalo.

- Yes
- No

8. Do you play videogames regularly? *

Marca solo un óvalo.

- Yes
- No

9. Have you ever used a virtual reality device? *

Marca solo un óvalo.

- Yes
- No

10. If yes, how often?

Marca solo un óvalo.

- Occasionally (5 times or less)
- Often
- Very often

11. If yes, check all that apply

Selecciona todos los que correspondan.

- [ ] I have tried computer-type devices (Oculus, HTC vive, Playstation VR)
- [ ] I have tried that use a smartphone
- [ ] I use VR regularly

12. If yes, have you ever experienced eyestrain, dizziness, headaches, or nausea in VR?

Marca solo un óvalo.

- Yes
- No

Este contenido no ha sido creado ni aprobado por Google.

Google Formularios

Appendix C. Sickness Questionnaire

Participants were asked to complete the questionnaire attached on the next page in order to see how the experiment affected possible vision-related symptoms that might appear. They had to fulfill it before and after the experiment. Only a few people reported experiencing some eyestrain and difficulty focusing after the experiment, but the majority of the participants reported experiencing none of the symptoms.

Sickness questionnaire

Questionnaire to assess whether the participants had experienced sickness or discomfort during the experiment

*Obligatorio

1. Subject ID *

2. Session *

Marca solo un óvalo.

- 1
- 2

3. Before or after the experiment? *

Marca solo un óvalo.

- Before
- After

4. Did you experience General Disconfort? *

Marca solo un óvalo.

	1	2	3	4	
No					A lot

5. Did you experience Fatigue? *

Marca solo un óvalo.

	1	2	3	4	
No					A lot

6. Did you experience Eyestrain? *

Marca solo un óvalo.

	1	2	3	4	
No					A lot

7. Did you experience Difficulty Focusing? *

Marca solo un óvalo.

	1	2	3	4	
No					A lot

8. Did you experience Headache? *

Marca solo un óvalo.

	1	2	3	4	
No					A lot

9. Did you experience Fullness of head? *

Marca solo un óvalo.

	1	2	3	4	
No					A lot

10. Did you experience Blurred Vision? *

Marca solo un óvalo.

	1	2	3	4	
No					A lot

11. Did you experience Dizzy? *

Marca solo un óvalo.

	1	2	3	4	
No					A lot

12. Did you experience Vertigo? *

Marca solo un óvalo.

	1	2	3	4	
No					A lot

13. Any other comments?

Este contenido no ha sido creado ni aprobado por Google.

Google Formularios

Appendix D. Presence Questionnaire

Finally, at the end of the experiment, participants were asked to complete the questionnaire attached on the next page in order to know their opinion about the experiment and their level of immersion while they were doing it. The answers collected are shown in the images below.

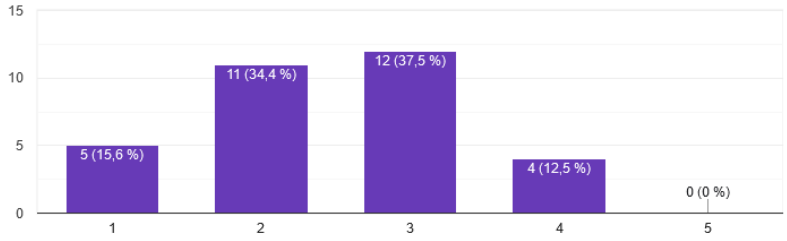


Figure D.1: Answers of participants to the question *How exciting was the experiment?*, being 1 none and 5 a lot.

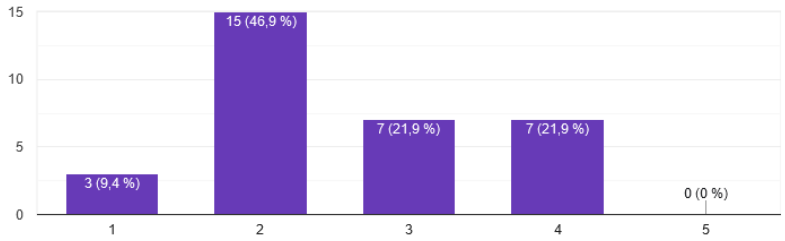


Figure D.2: Answers of participants to the question *Did you feel present in the virtual environment?*, being 1 none and 5 a lot.

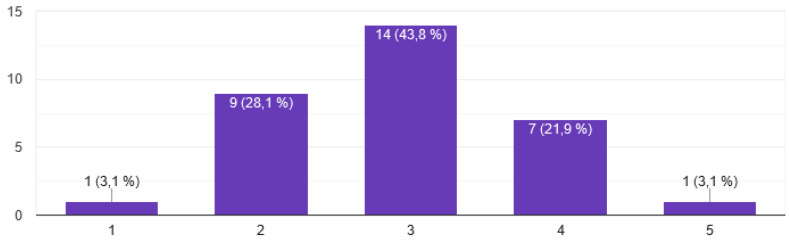


Figure D.3: Answers of participants to the question *How aware were you of the real world surrounding while doing the experiment?*, being 1 completely unaware and 5 extremely aware.

Presence questionnaire

Questionnaire to know the participants' opinion about the experiment.

*Obligatorio

1. Subject ID *

2. How exciting was the animation? *

Marca solo un óvalo.

	1	2	3	4	5	
Not exciting at all						Very exciting

3. In the computer generated world I had a sense of "being there" *

Marca solo un óvalo.

	1	2	3	4	5	
Nothing at all						A lot

4. I felt present in the virtual space *

Marca solo un óvalo.

	1	2	3	4	5	
Nothing at all						A lot

5. How aware were you of the real world surrounding while experiencing the virtual world? (i.e. sounds, room temperature, other people, etc.) *

Marca solo un óvalo.

	1	2	3	4	5	
I was completely unaware of what was going on around me						I was extremely aware of what was going on around me

6. Any other comments?

Este contenido no ha sido creado ni aprobado por Google.

Google Formularios

Appendix E. Experiment details

This appendix complements the chapter of the experiment (Section 3), giving more detailed information on how the figures were normalized and showing the meshes used in the experiment.

E.1 Meshes normalization

An example to illustrate how the shapes were normalized is shown below. Let's imagine that there is a cube centered at the point $(2, 2, 2)$ of side 2 (blue cube in Fig. E.1). This means that the coordinates of its eight vertices are $(1, 1, 1)$, $(3, 1, 1)$, $(3, 3, 1)$, $(1, 3, 1)$, $(1, 1, 3)$, $(1, 3, 3)$, $(3, 1, 3)$ and $(3, 3, 3)$. The midpoint of these vertices (the centroid of the cube) is $(2, 2, 2)$, and it is computed as shown in Eq. E.1, where N is the number of vertices that make up the shape and (x_i, y_i, z_i) are the coordinates of the i^{th} vertex. If this centroid is deduced from the original cube, the new cube obtained is already centered at the origin and its vertices are $(-1, -1, -1)$, $(1, -1, -1)$, $(1, 1, -1)$, $(-1, 1, -1)$, $(-1, -1, 1)$, $(-1, 1, 1)$, $(1, -1, 1)$ and $(1, 1, 1)$.

$$x_{av} = \frac{\sum_{i=1}^{N} x_i}{N} \qquad y_{av} = \frac{\sum_{i=1}^{N} y_i}{N} \qquad z_{av} = \frac{\sum_{i=1}^{N} z_i}{N} \tag{E.1}$$

Now, to scale the cube, the norms of all these vertices are computed. In this case, all norms have a value of 1.73 because all their coordinates have a value of 1 or -1, and squaring them always returns 1. So, the maximum is 1.73, and this is the scale factor. Eq. E.2 shows an example to compute the norm of a specific vertex.

$$||(-1, -1, 1)|| = \sqrt{(-1)^2 + (-1)^2 + 1^2} = \sqrt{3} = 1.73 \tag{E.2}$$

Finally, dividing the coordinates of the vertices of the cube by this scale factor, the resulting cube, red cube in Fig. E.1, has its vertices at $(-0.6, -0.6, -0.6)$, $(0.6, -0.6, -0.6)$, $(0.6, 0.6, -0.6)$, $(-0.6, 0.6, -0.6)$, $(-0.6, -0.6, 0.6)$, $(-0.6, 0.6, 0.6)$, $(0.6, -0.6, 0.6)$ and $(0.6, 0.6, 0.6)$. This process was done for all the meshes so that they have the same size, position and orientation.

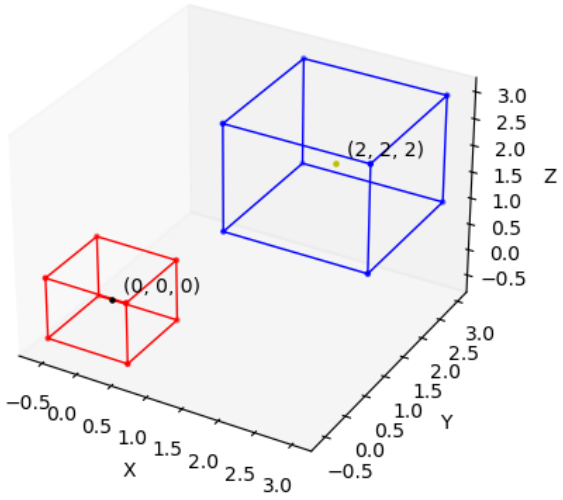


Figure E.1: Example of scaling and centering a mesh at the origin.

E.2 Meshes used in the experiment

All meshes used in the experiment are shown below.

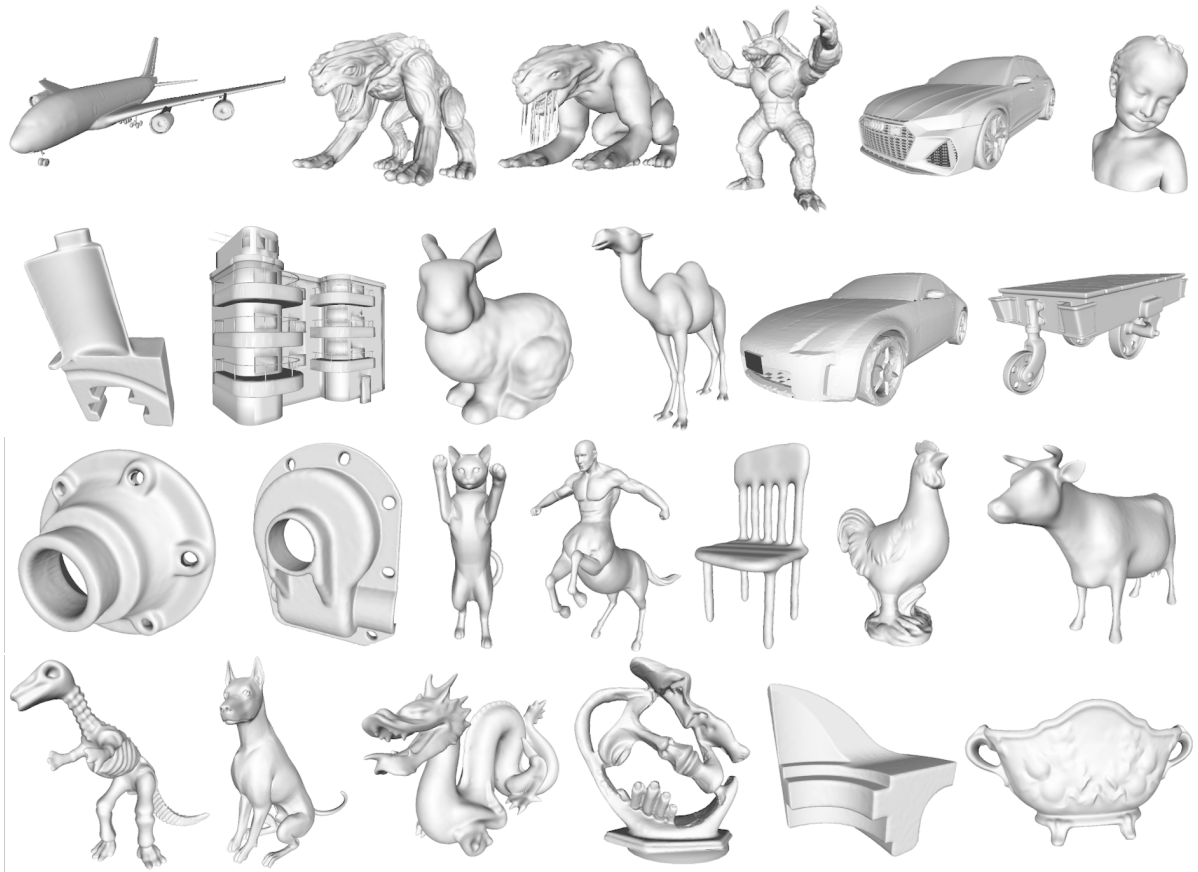


Figure E.2: Meshes used for the experiment (1).

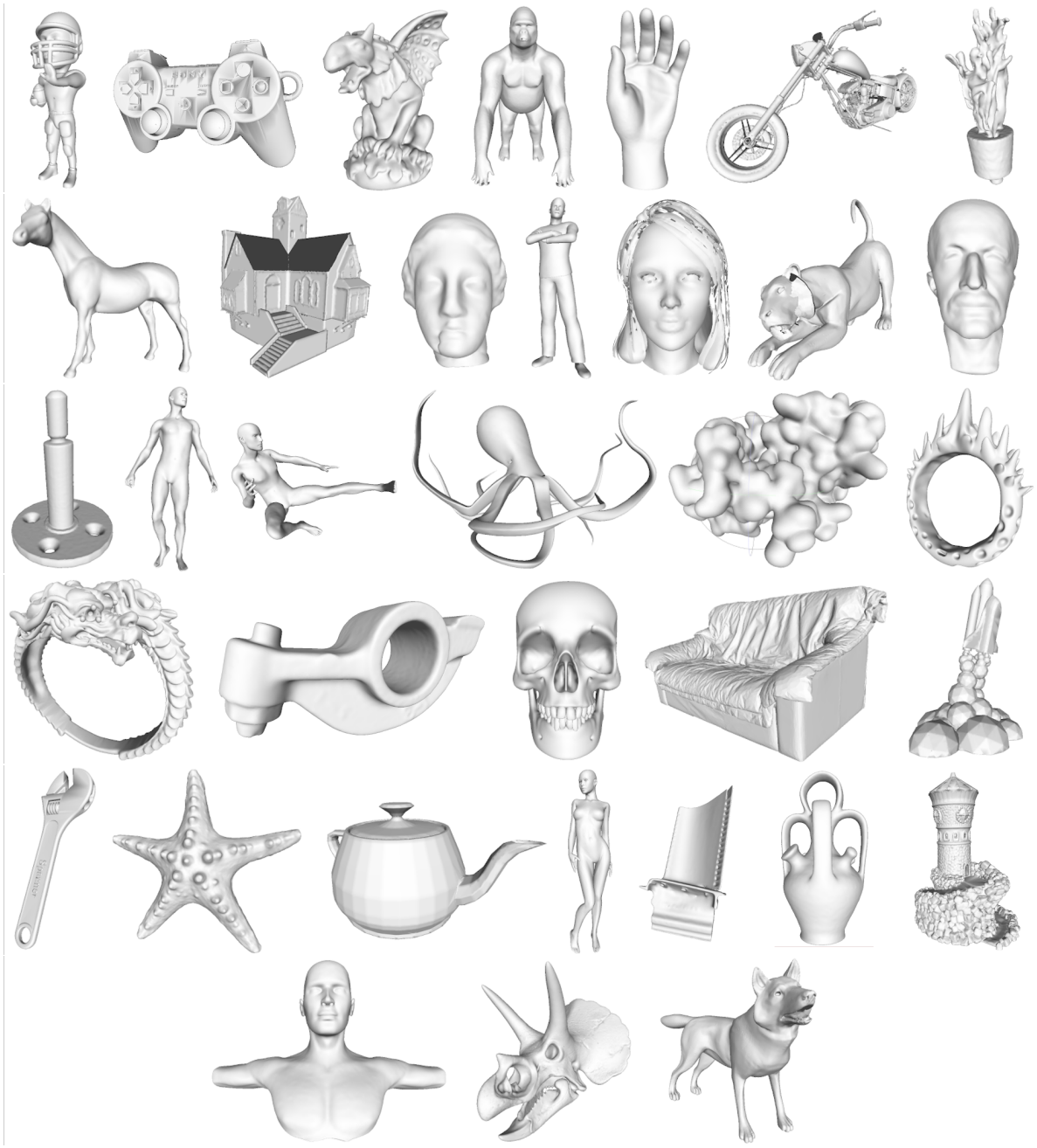


Figure E.3: Meshes used for the experiment (2).

Appendix F. Point clouds smoothing

The smoothing process carried out on the point clouds in order to create a saliency map for each shape from the raw fixation maps has been as follows:

1. Take vertex by vertex.
2. Check if the fixation map corresponding to that vertex is 0.
3. If its value is 0, move on to the next vertex.
4. If its value is not zero, take the k nearest neighbors and create a negative linear function, i.e. the nearest vertex (itself) is assigned its own value, and the farthest vertex is assigned a value of 0.
5. When this process has been done for all vertices, move on to the next shape and repeat the process.

As an example, let's imagine that a vertex has a fixation map of value 1.3 and its farthest neighbour is 0.75cm from it. Then, the closest vertex (itself) would be assigned 1.3 and the farthest vertex would be assigned 0. Fig.F.1 shows the function of this negative linear relation and Fig. F.2 shows the resulting point clouds after the smoothing.

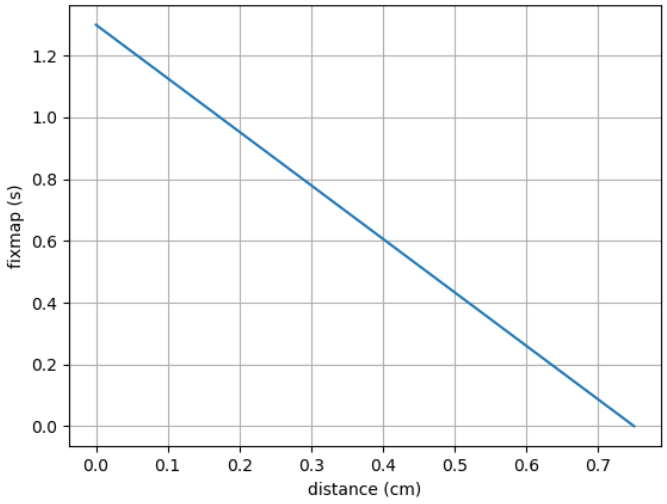


Figure F.1: Negative linear relation applied for the smoothing.

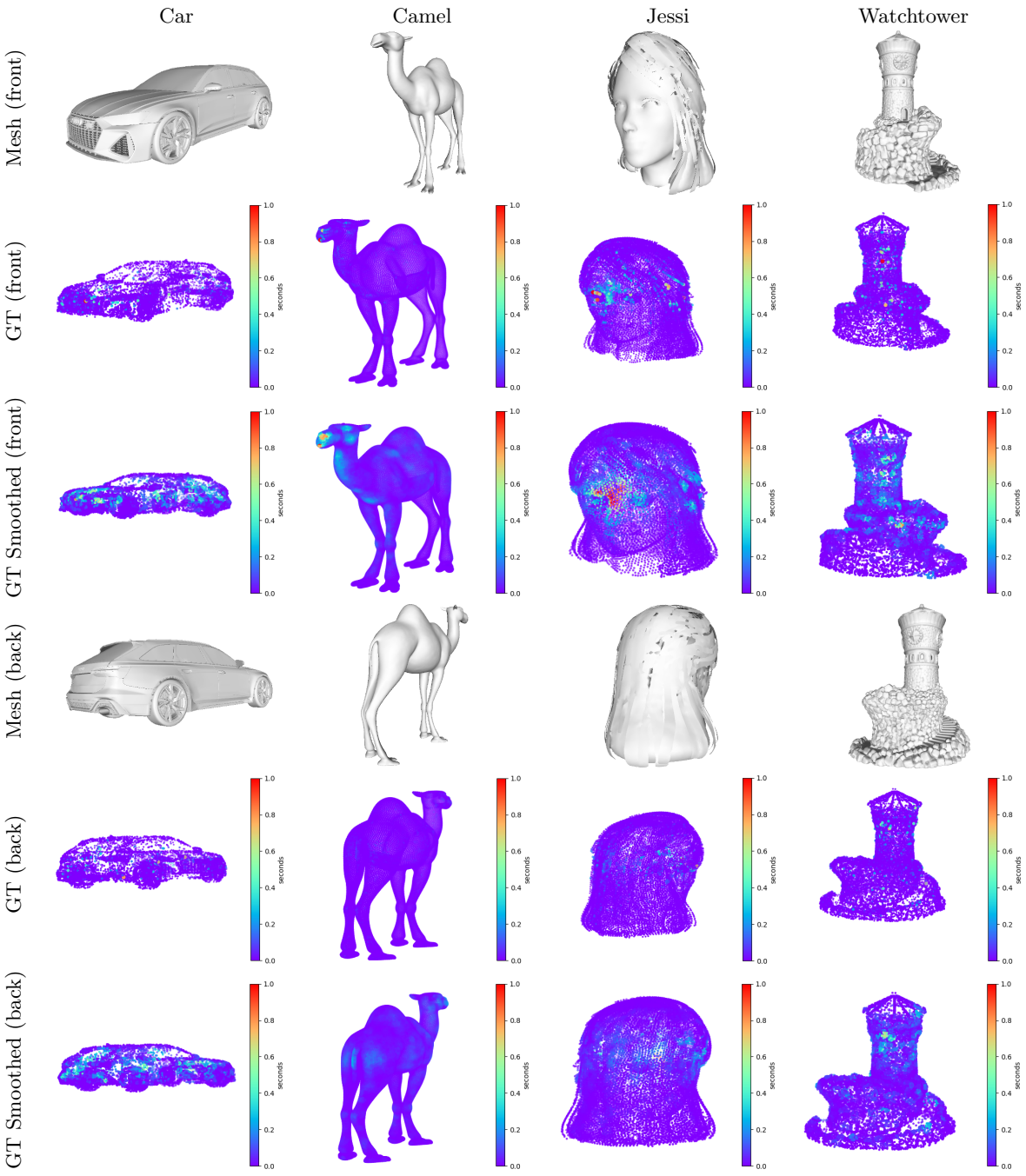


Figure F.2: Resulting point clouds with the smoothing.

Appendix G. Quantitative evaluation

This appendix presents the detailed results obtained in the quantitative evaluations performed in the ablation study (Section 5.4). The scores obtained with the metrics mentioned in Section 5.1 are presented in this appendix by point cloud, instead of averaged.

G.1 Input resolutions

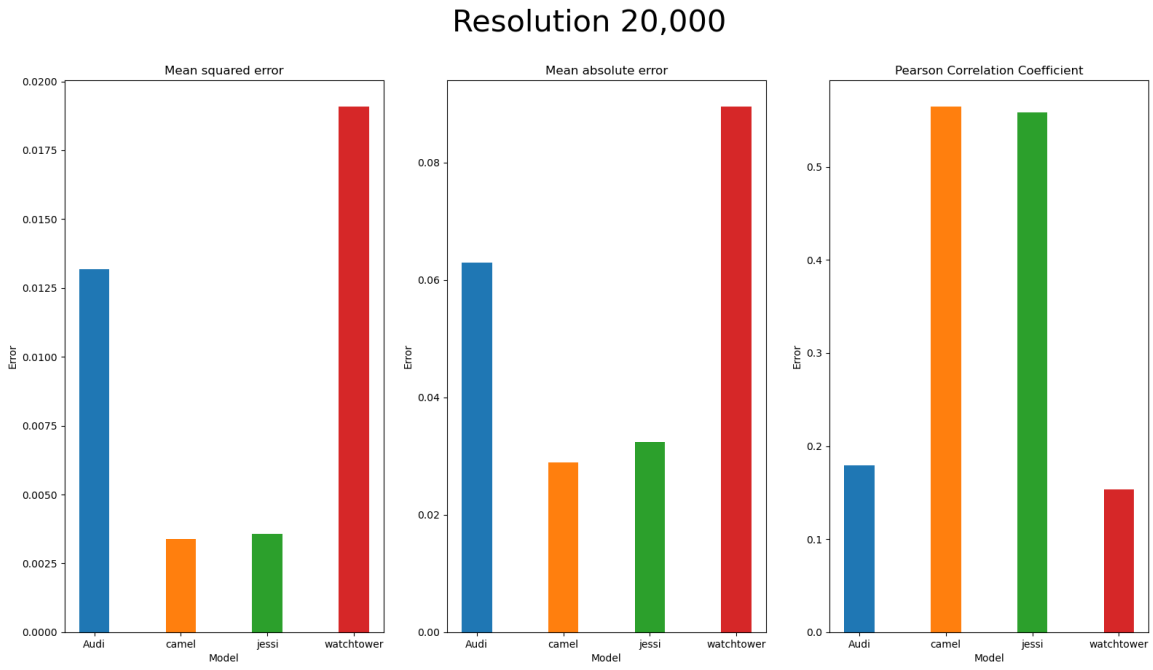


Figure G.1: Individual metrics for the resolution of 20,000 vertices taking into account all fixation maps.

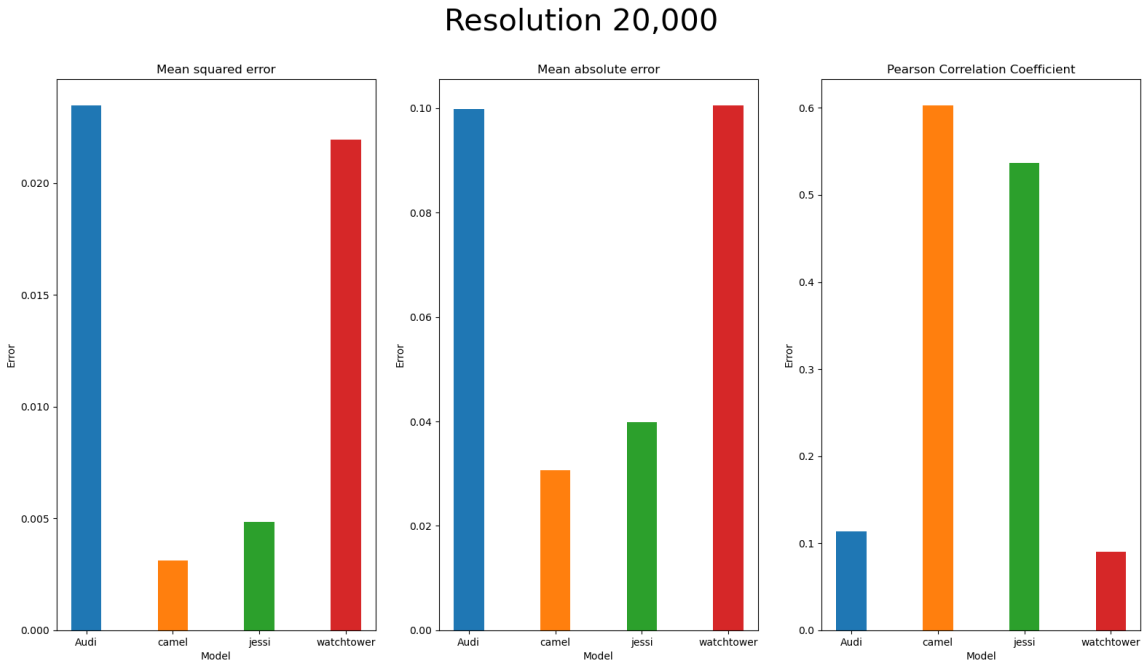


Figure G.2: Individual metrics for the resolution of 20,000 vertices taking into account only the non-zero fixation maps.

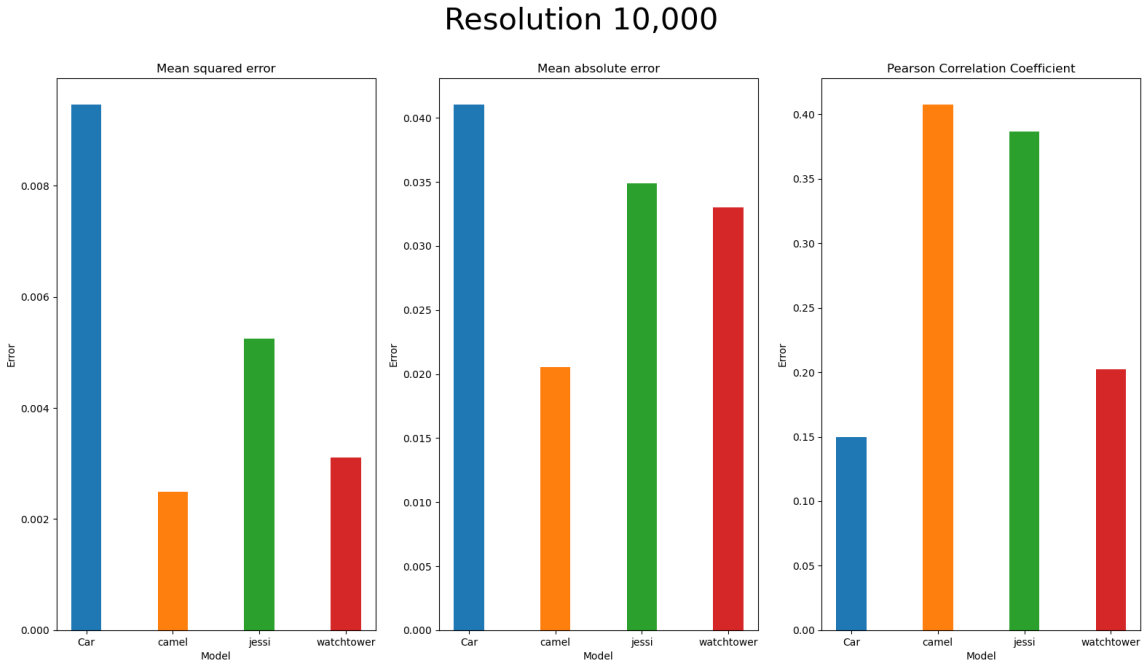


Figure G.3: Individual metrics for the resolution of 10,000 vertices taking into account all fixation maps.

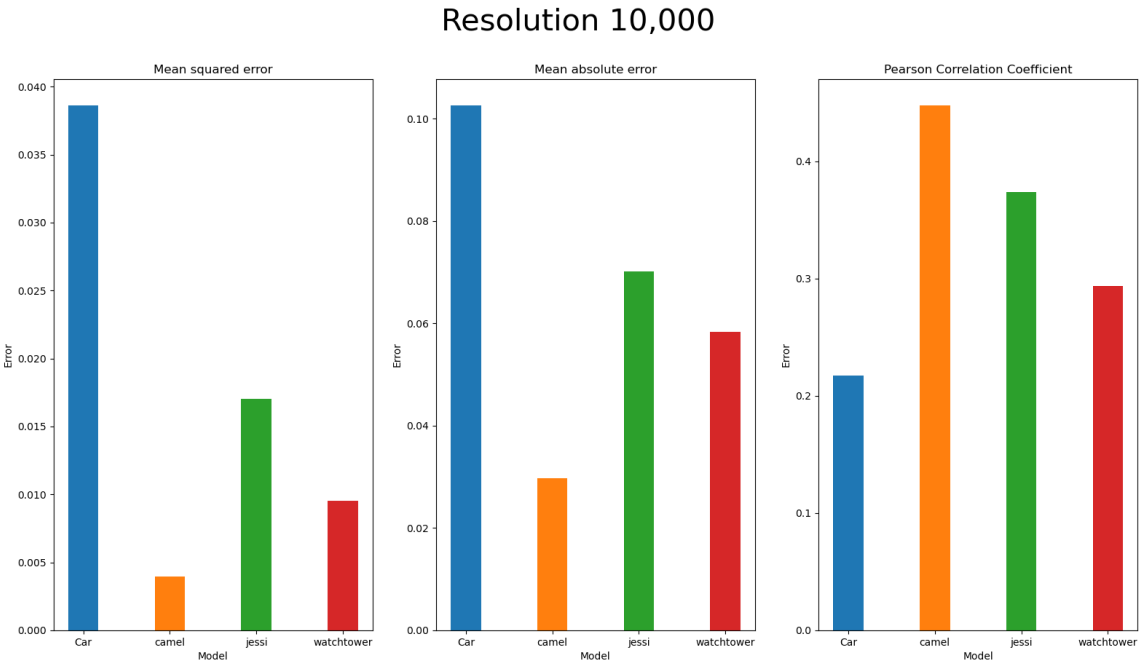


Figure G.4: Individual metrics for the resolution of 10,000 vertices taking into account only the non-zero fixation maps.

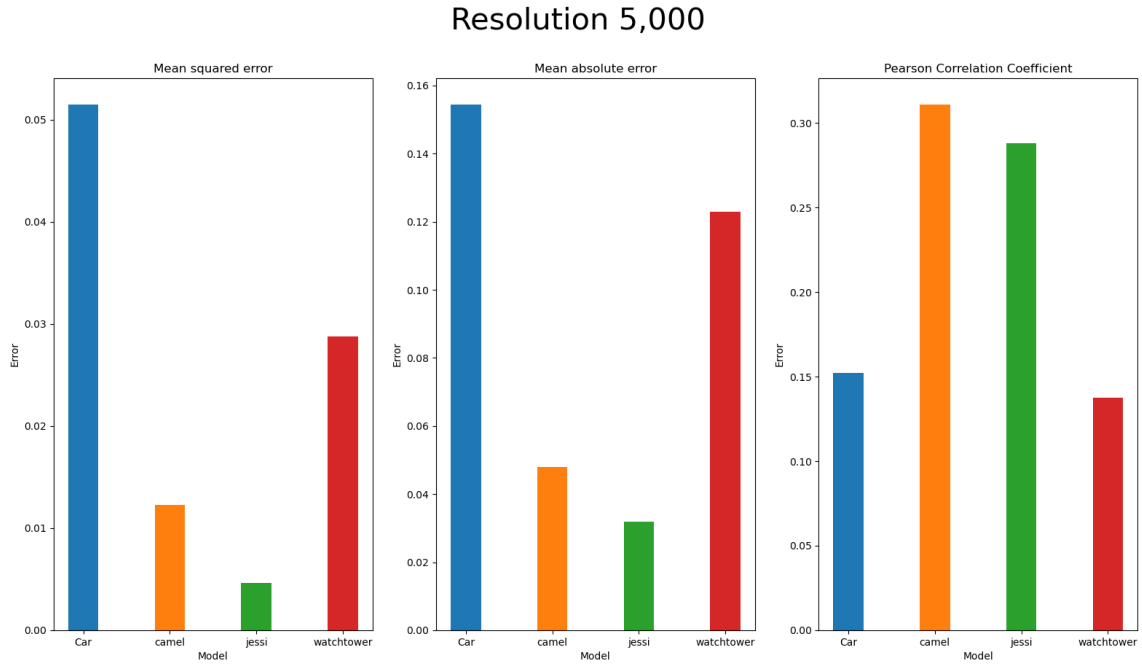


Figure G.5: Individual metrics for the resolution of 5,000 vertices taking into account all fixation maps.

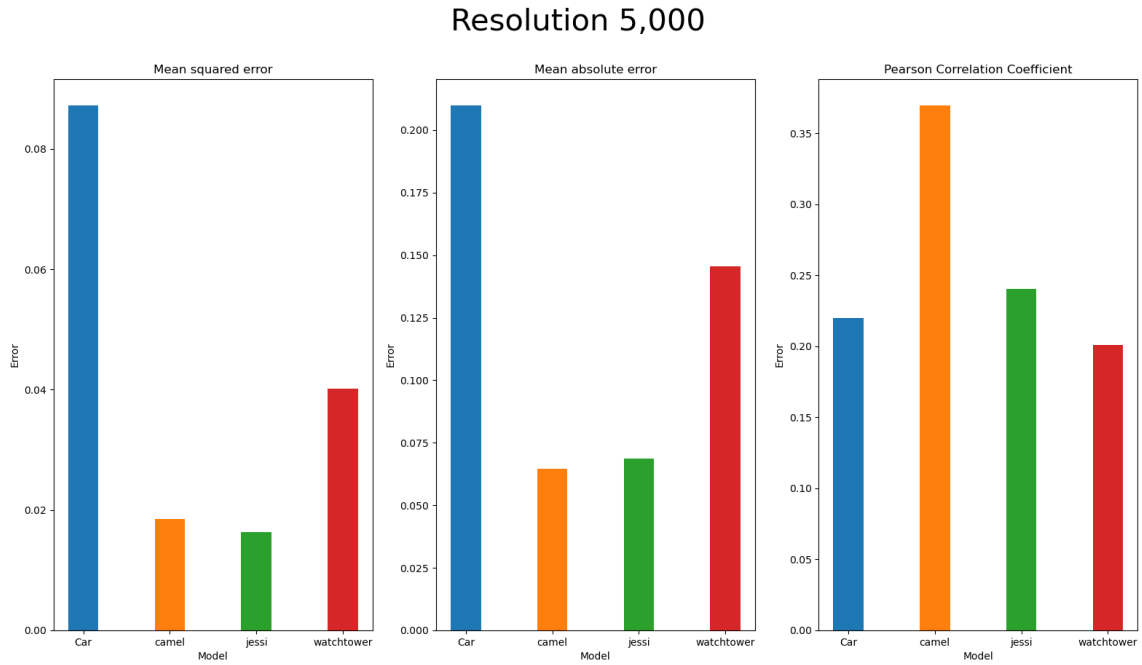


Figure G.6: Individual metrics for the resolution of 5,000 vertices taking into account only the non-zero fixation maps.

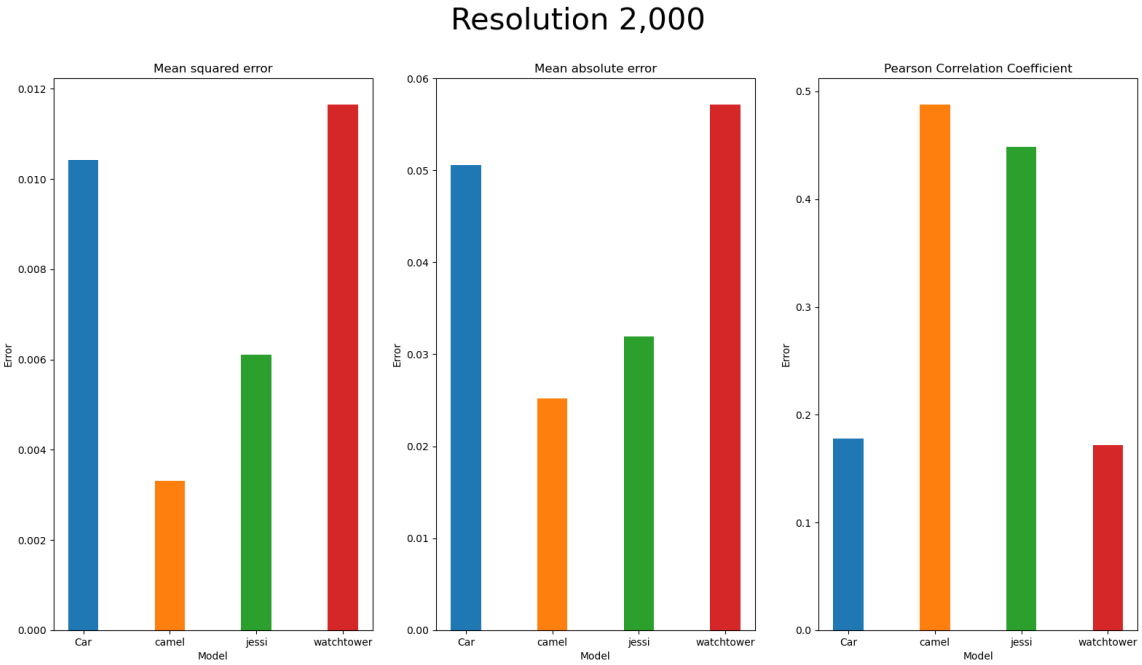


Figure G.7: Individual metrics for the resolution of 2,000 vertices taking into account all fixation maps.

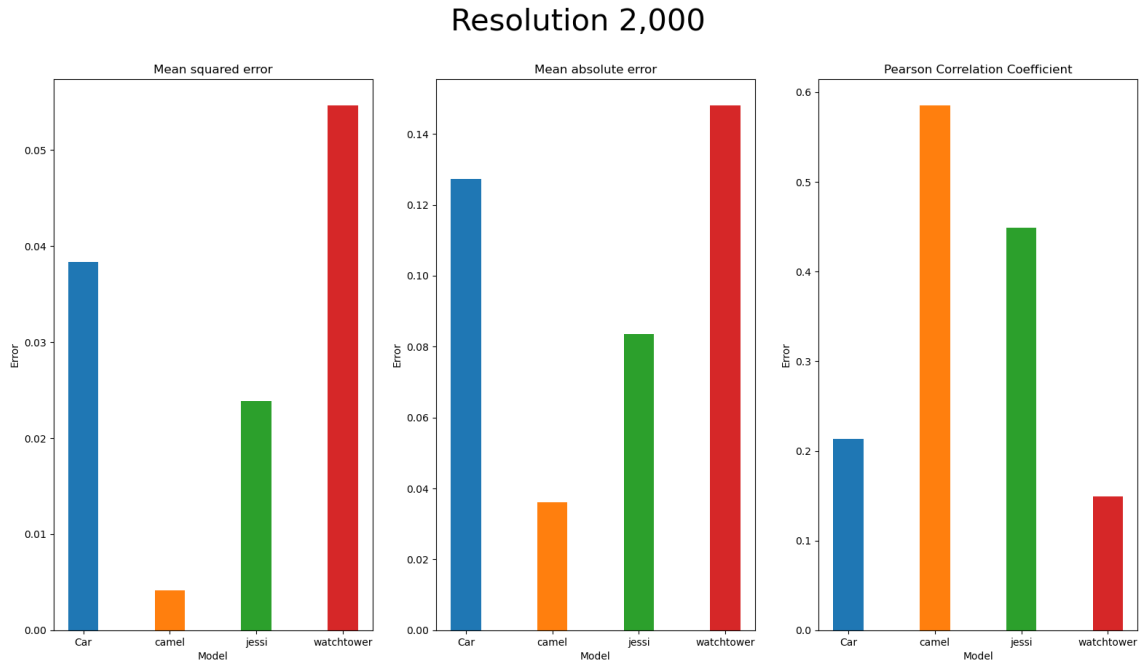


Figure G.8: Individual metrics for the resolution of 2,000 vertices taking into account only the non-zero fixation maps.

G.2 Loss functions

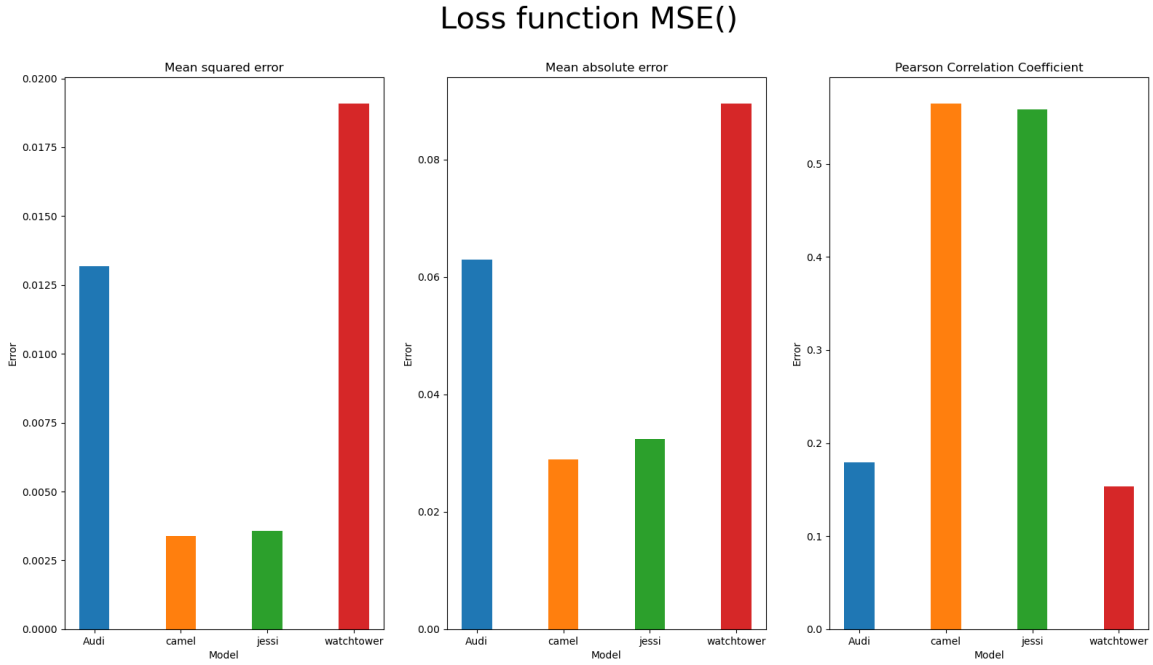


Figure G.9: Individual metrics for the MSE loss function taking into account all fixation maps.

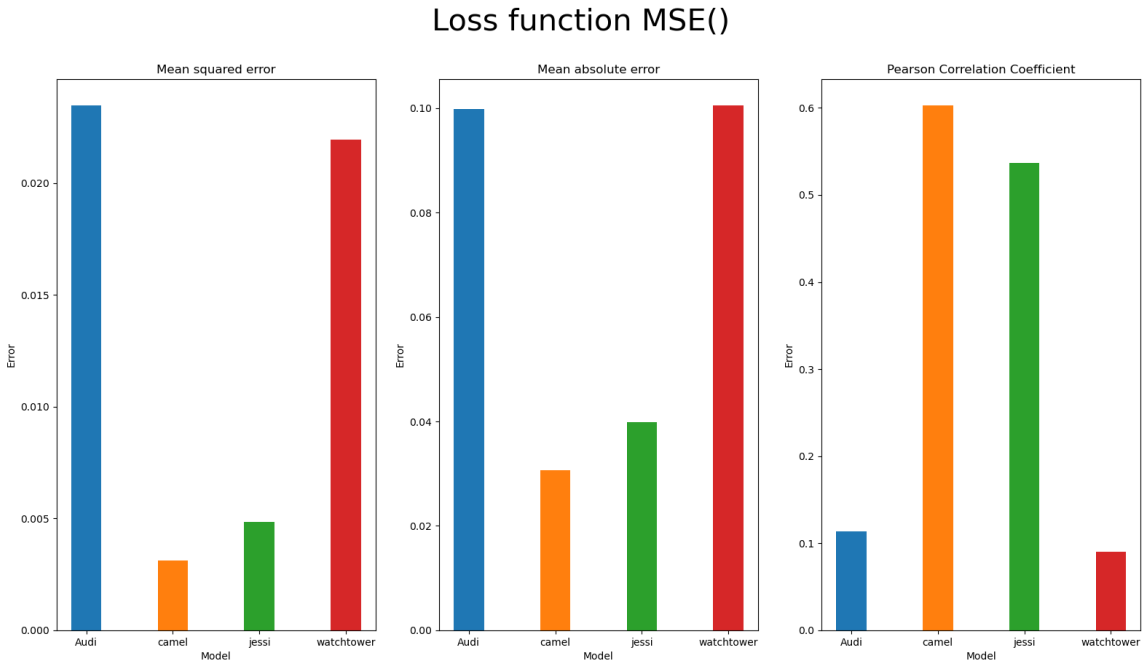


Figure G.10: Individual metrics for the MSE loss function taking into account only the non-zero fixation maps.

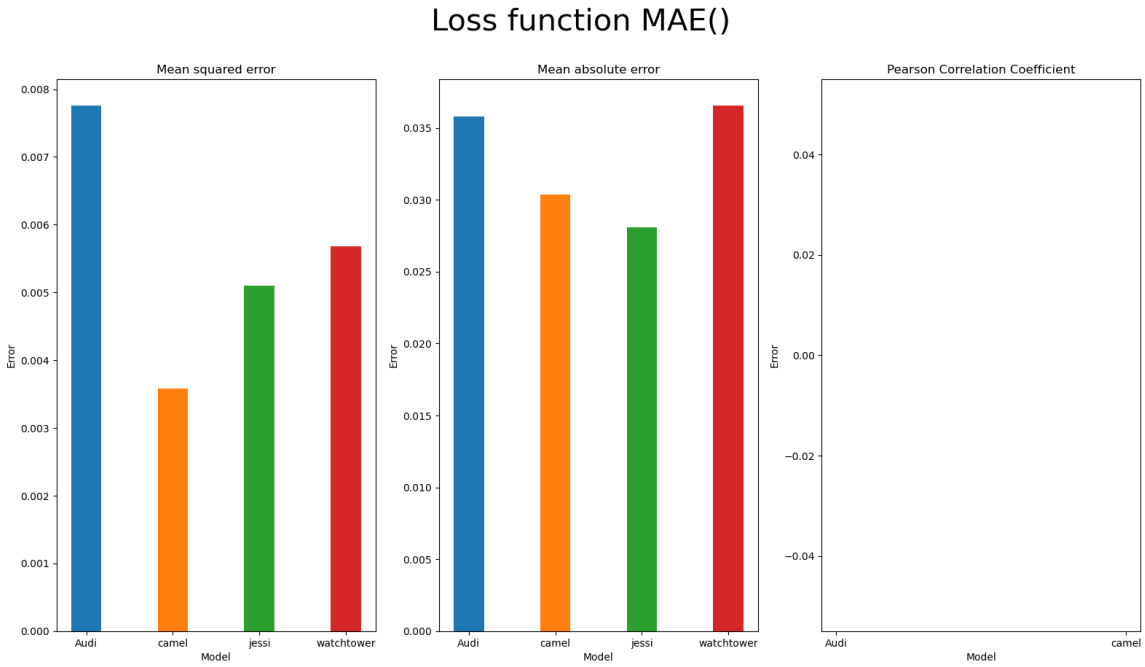


Figure G.11: Individual metrics for the MAE loss function taking into account all fixation maps.

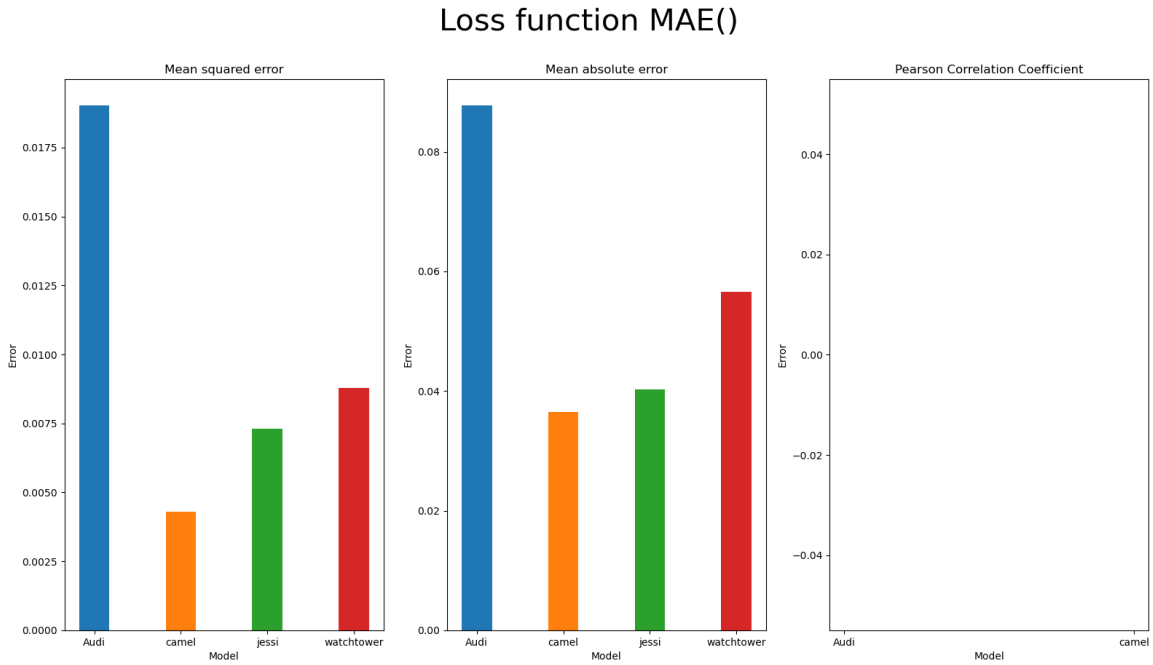


Figure G.12: Individual metrics for the MAE loss function taking into account only the non-zero fixation maps.

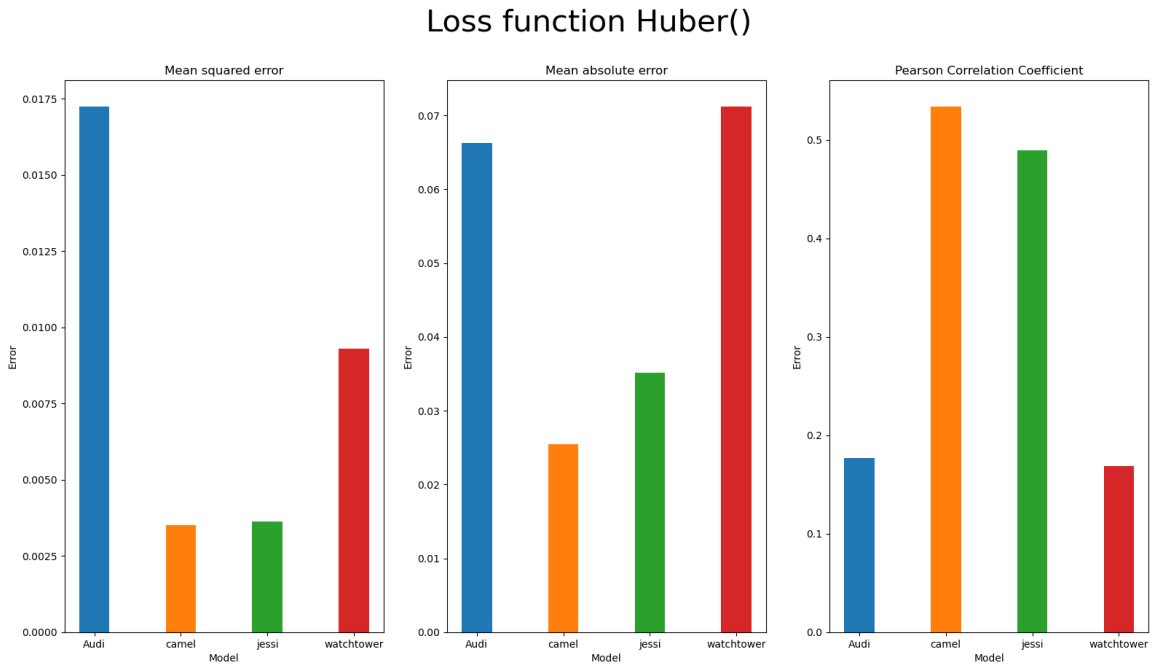


Figure G.13: Individual metrics for the Huber loss function taking into account all fixation maps.

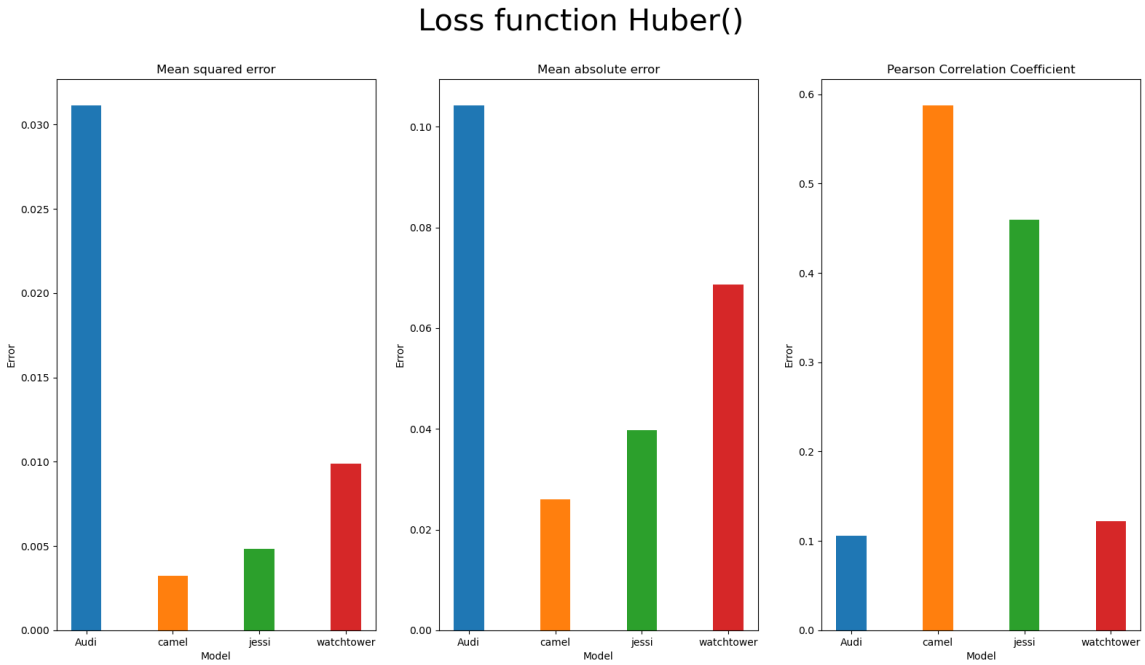


Figure G.14: Individual metrics for the Huber loss function taking into account only the non-zero fixation maps.

G.3 Architecture variations

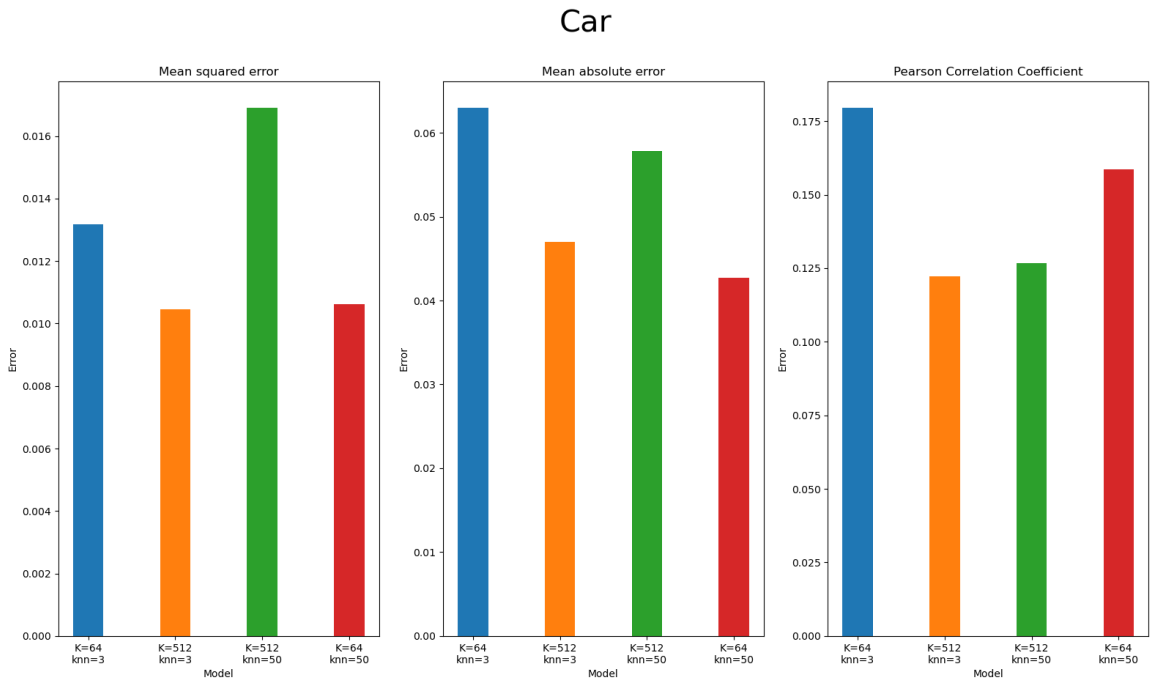


Figure G.15: Individual metrics for the car taking into account all fixation maps.

Figure G.16: Individual metrics for the car taking into account only the non-zero fixation maps.

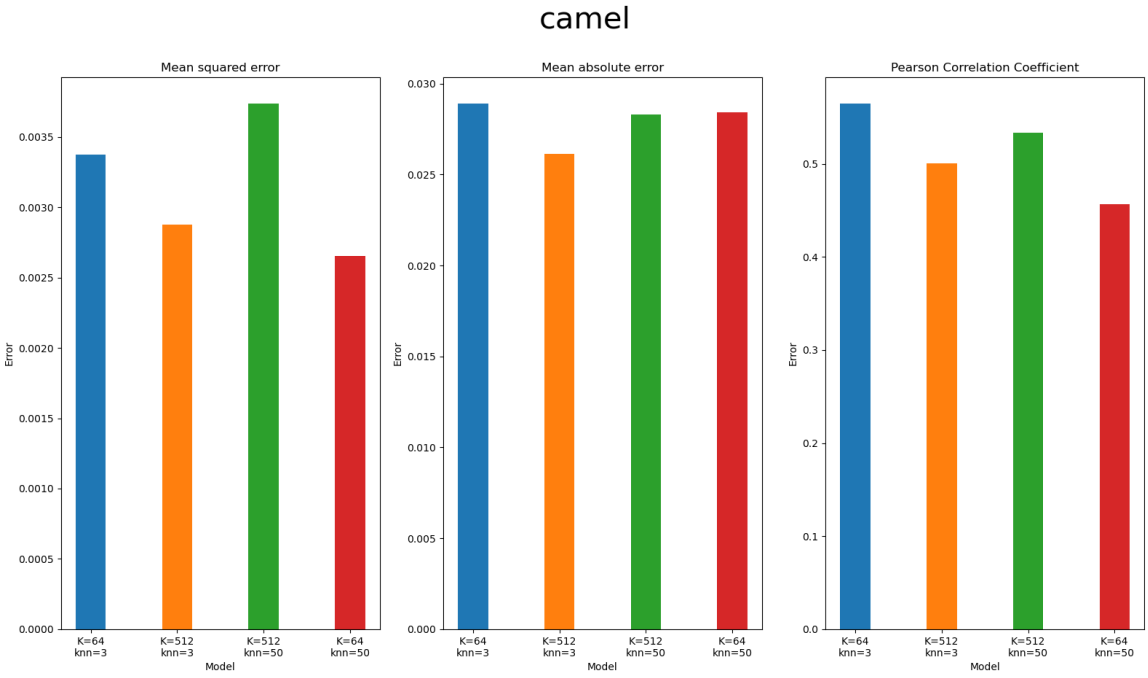


Figure G.17: Individual metrics for the camel taking into account all fixation maps.

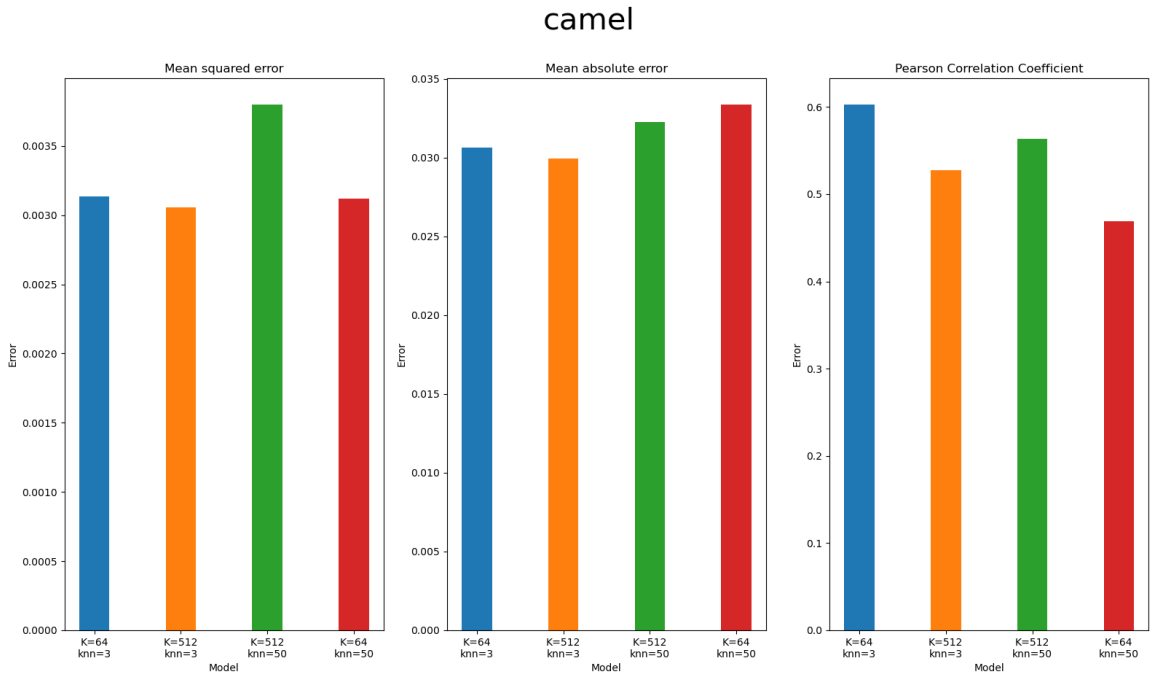


Figure G.18: Individual metrics for the camel taking into account only the non-zero fixation maps.

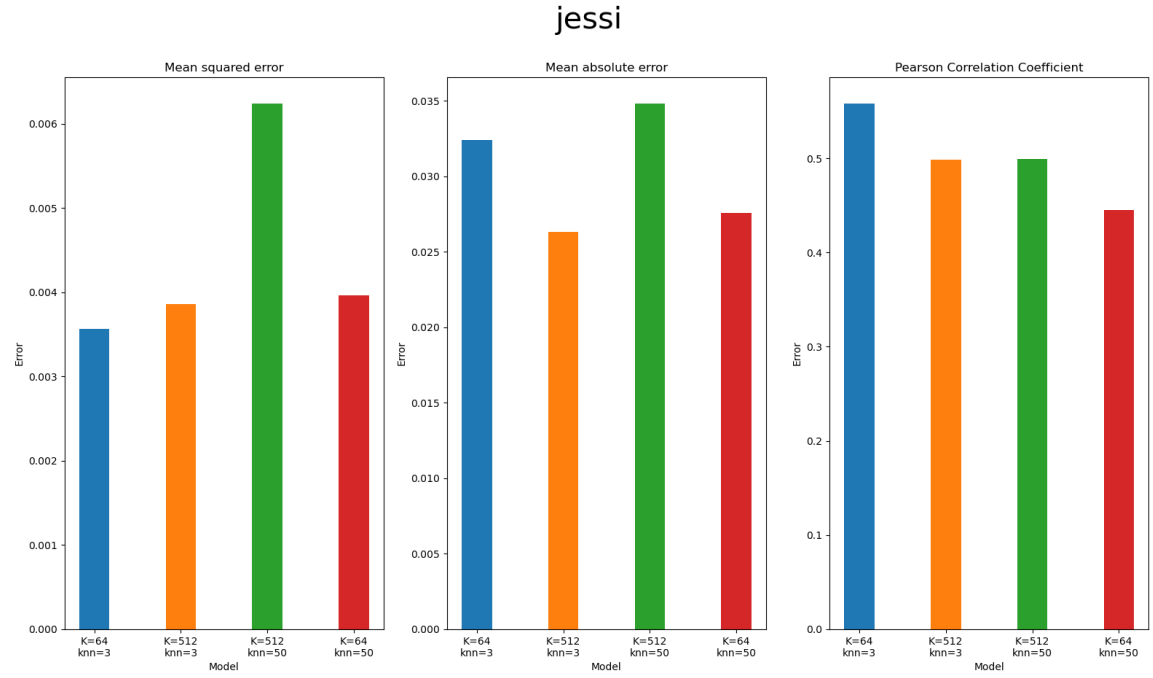


Figure G.19: Individual metrics for the girl taking into account all fixation maps.

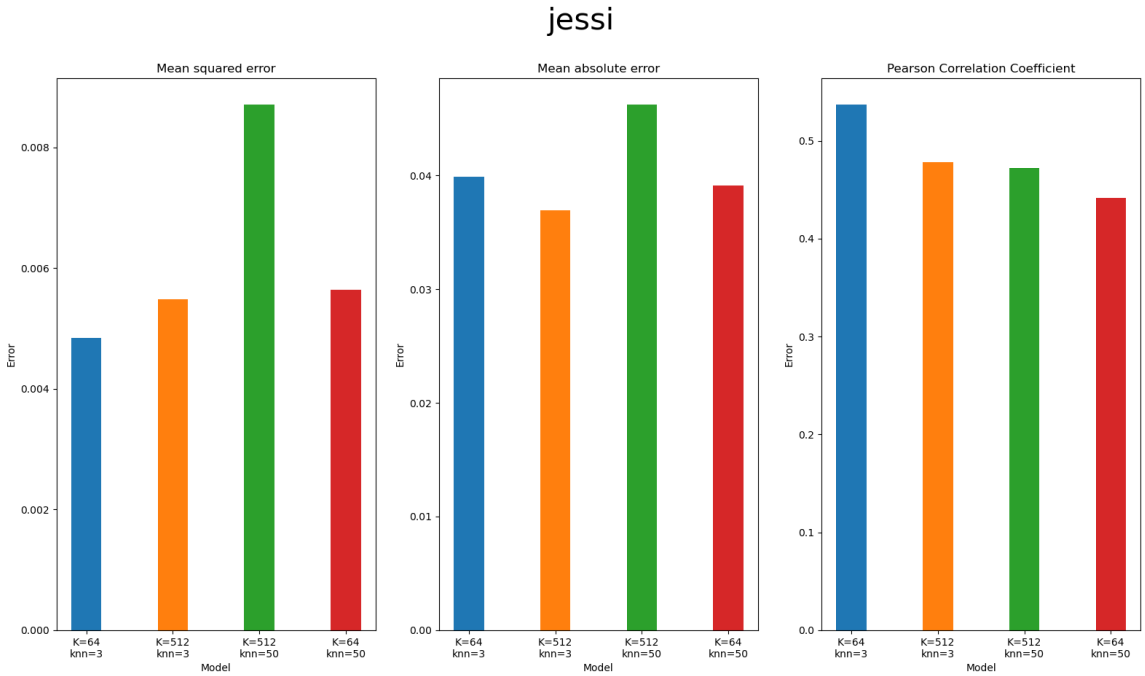


Figure G.20: Individual metrics for the girl taking into account only the non-zero fixation maps.

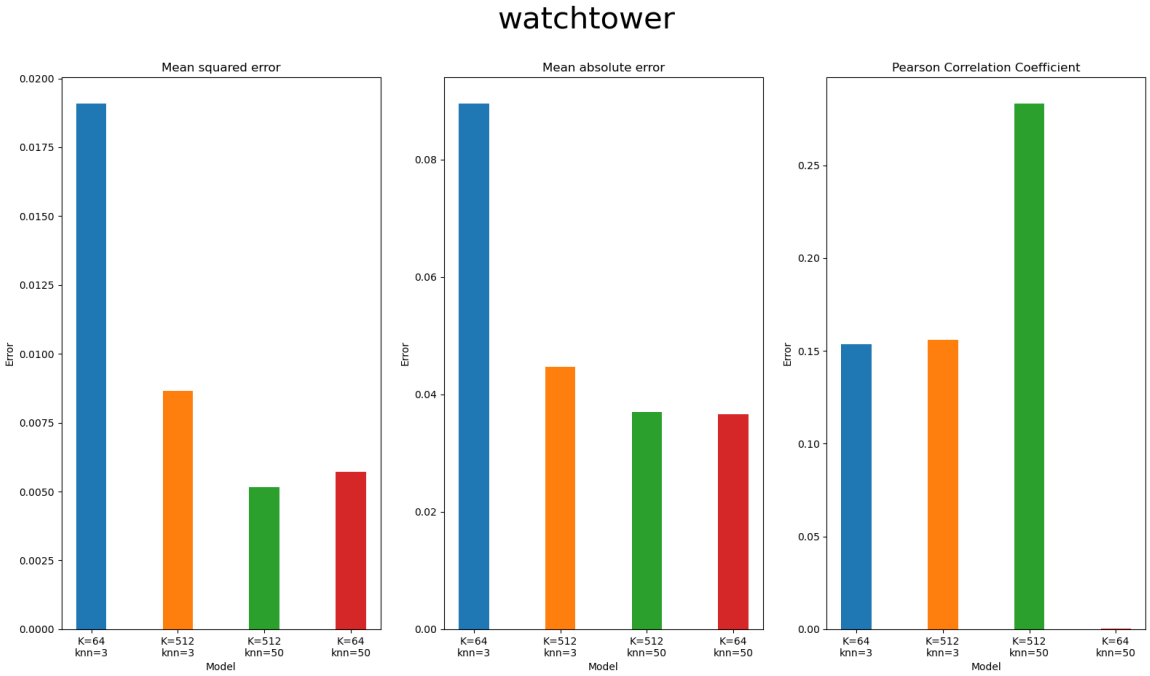


Figure G.21: Individual metrics for the watchtower taking into account all fixation maps.

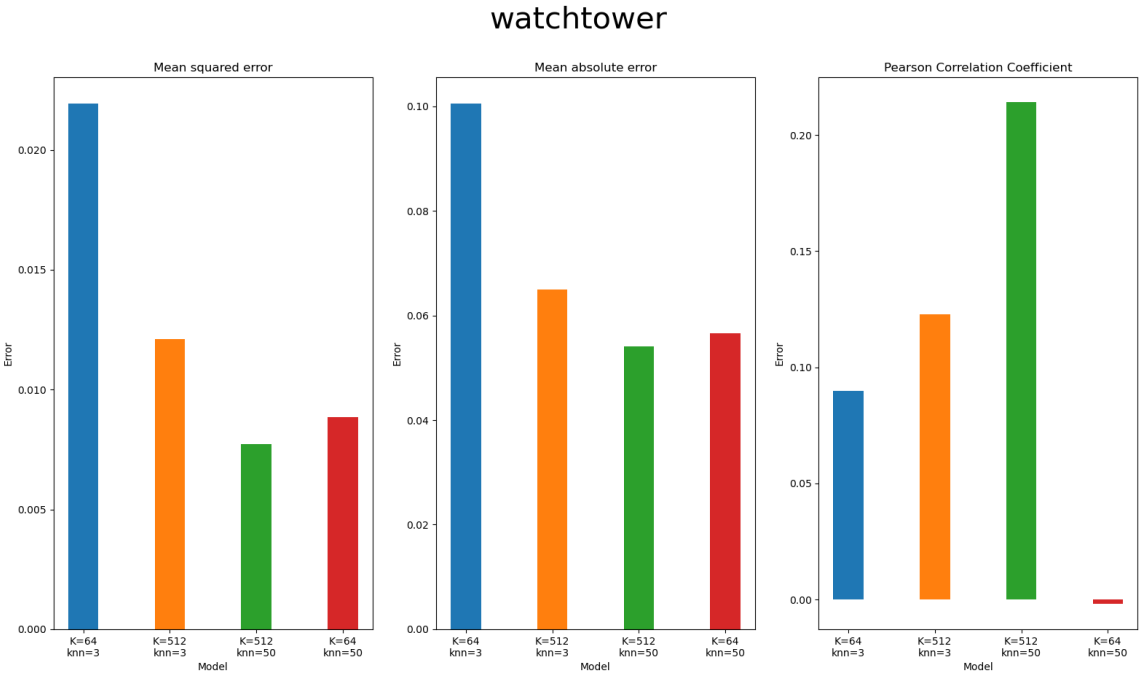


Figure G.22: Individual metrics for the watchtower taking into account only the non-zero fixation maps.