Universidad Zaragoza
1542

Master's Thesis

Audio-visual saliency prediction for 360° video via deep learning

Author

Félix Bernal Sierra

Supervisors

Belén Masiá Corcoy

Ana Serrano Pacheu

Master in Robotics, Graphics and Computer Vision
Escuela de Ingeniería y Arquitectura
2022

Abstract

The interest in virtual reality (VR) has rapidly grown in recent years, being now widely available to consumers in different forms. This technology provides an unprecedented level of immersion, creating many new possibilities that could change the way people experience digital content. Understanding how users behave and interact with virtual experiences could be decisive for many different applications such as designing better virtual experiences, advanced compression techniques, or medical diagnosis.

One of the most critical areas in the study of human behaviour is visual attention. It refers to to the qualities that different items have which makes them stand out and attract our attention. Despite the fact that there have been significant advances in this field in recent years, saliency prediction remains a very challenging problem due to the many factors that affect the behavior of the observer, such as stimuli sources of different types or users having different backgrounds and emotional states. On top of that, saliency prediction for VR content is even more difficult as this form of media presents additional challenges such as distortions, users having control of the camera, or different stimuli possibly being located outside the current view of the observer.

This work proposes a novel saliency prediction solution for 360° video based on deep learning. Deep learning has been proven to obtain outstanding results in many different image and video tasks, including saliency prediction. Although most works in this field focus solely on visual information, the proposed model incorporates both visual and directional audio information with the objective of obtaining more accurate predictions. It uses a series of convolutional neural networks (CNNs) specially designed for VR content, and it is able to learn spatio-temporal visual and auditory features by using three-dimensional convolutions. It is the first solution to make use of directional audio without the need for a hand-crafted attention modelling technique.

The proposed model is evaluated using a publicly available dataset. The results show that it outperforms previous state-of-the-art work in both quantitative and qualitative analysis. Additionally, various ablation studies are presented, supporting the decisions made during the design phase of the model.

Acknowledgments

I would like to thank my supervisors, Belén Masiá and Ana Serrano, for giving me the opportunity to work with them on such an interesting topic. Thank you for the help and support you have given me throughout these months.

Contents

List of Figures

List of Tables

1. Introduction

1.1 Context

In recent years, there has been a rapidly growing interest in virtual reality (VR). This emerging technology is no longer limited to laboratories or research facilities; it is now widely accessible to consumers through different devices, with the most common one being head-mounted displays (HMD). These devices give the user the feeling of being present in a virtual world by a simulating stereoscopic vision, allowing the user to perceive depth by showing different images to each eye, and tracking the movement of the head of the user, matching it with the content presented within the virtual world.

There are many different VR applications currently available, such as training and education, entertainment, or treating conditions including anxiety and phobias. Despite that, this new technology still poses challenges and constraints that remain unsolved. Among others, traditional media is typically consumed on a screen, whereas in VR the user can explore the environment with different degrees of freedom (DOF) (see Figure 1.1). In VR, the user controls the orientation of the camera and only a portion of the scene is seen at once. This supposes a paradigm shift that makes some previous works targeted towards traditional forms of media not be applicable for VR, and thus new studies are needed.

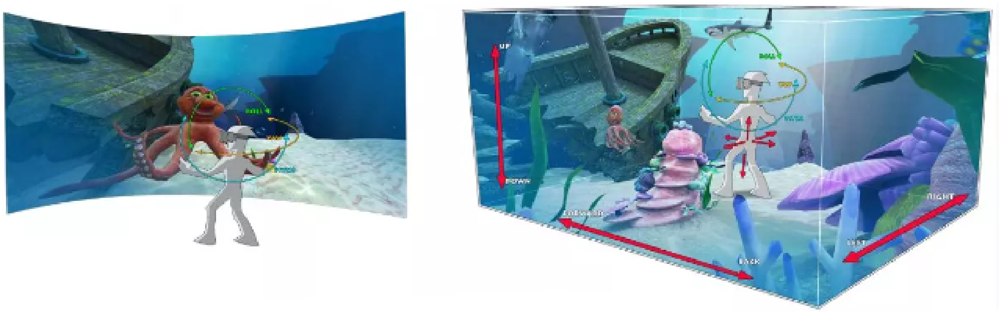

Figure 1.1: 3-DOF (left) and 6-DOF (right) content. In 6-DOF content, the user can move around and explore the virtual scenario; whereas in 3-DOF content, the user cannot move but only look around in different directions.

Research on the subject of VR has been rapidly growing due to the many new possibilities that it creates and its expanding availability. One key area that has drawn particular attention of many researchers is understanding how users behave in VR environments. This could help create more engaging experiences and better applications overall, as well as improve key aspects of this technology such as user interfaces or tracking technologies, which in turn would help to develop VR technologies even further.

Within the study of human behavior, a critical part is visual attention, which refers to how different stimuli affect the eye movements of the observer. These movements are typically divided into fixations, when the gaze is fixed on a particular point; and saccades, which are rapid movements that occur between different fixations. One of the most frequent techniques used to model visual behaviour is saliency (see Figure 1.2), which models the probability of each element (pixel or object) in the image to attract the attention of the observer (establish a fixation).

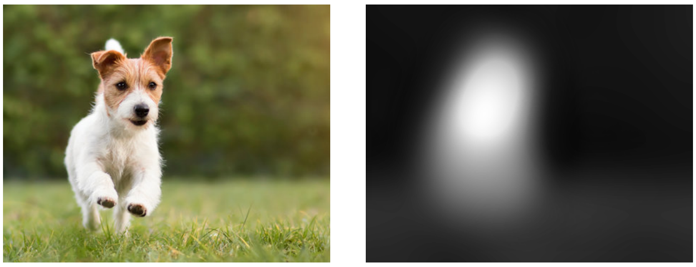

Figure 1.2: Example of an image (left) and its saliency map (right). The saliency map provides a value for each pixel through a grayscale image that represents the probability that the observer establishes a fixation on that position. Black values mean low probability and white values mean high probability.

In computer vision, the problem of modelling and predicting visual saliency has long been studied [4, 5, 6, 7], with recent works based on deep learning rendering previous ones obsolete [8, 9]. As mentioned above, VR presents numerous challenges that make this and other tasks particularly difficult. Thus, works targeted specially towards VR have been developed.

Many different approaches have been developed to predict saliency in 360° images, some of them with very promising results[10, 11, 12, 13]; helping us to better understand what factors drive certain behaviors. In the case of 360° videos, the addition of motion makes the task much more complex. Multiple works have incorporated some kind of temporal information to improve their predictions on video sequences [12, 14, 15]. Most of these works take into account only what is seen by the observer. However, different works [16, 17, 18] have shown that auditory cues can have a significant impact on visual attention. Audio can help orient users in VR and influence their behavior, which could be crucial, e.g., when the source of the audio is outside the view of the observer. For this reason, it is important to consider both visual and auditory information when modelling VR behavior.

To the best of our knowledge, the work by Chao et al.[3] proposes the only solution for saliency prediction in 360° videos that uses visual and auditory information to obtain its predictions. It incorporates directional audio by utilising Audio Energy Maps (AEMs) (Section 3.1.1), which represent the intensity of the audio coming from different directions. This information is used to successfully improve its predictions and obtain more accurate results through an attention modelling mechanism.

The objective of this master's thesis is to propose a novel saliency prediction model for 360° videos that utilises both visual and directional audio information without making use of a manually defined attention modelling technique. By using auditory information, the model could possibly predict more accurately what parts of the scene are salient on different types of scenes. Furthermore, spatial auditory information could give the model the ability to further refine its predictions towards the sound sources. The novelty of the problem, as well as the many factors that have an effect on visual attention in 360° videos, make the task of saliency prediction in this new format extremely challenging. This can be observed in the limited amount of works focused on this subject.

The model proposed in this work, based on deep learning, incorporates directional audio information through AEMs (Section 3.1.1) and uses a special type of convolutional neural network (CNN) tailored specifically for 360° content (Section 4.2). It outperforms the previous state-of-the-art work, which also incorporates directional audio information [3], in both quantitative and qualitative results. The evaluation is performed on a dataset composed of 67 videos with different real-world scenes. In addition to that, an ablation study is presented that supports the decisions made during the design phase of the model.

1.2 Objectives and scope

The objective of this work is to design, implement and evaluate a solution to predict saliency in 360° video, taking into account both visual and directional audio information. This is achieved through specific objectives:

- Study of the state of the art in saliency prediction in traditional and 360° content, for both static images and video, with and without audio information (Section 2).
- Design and implementation of an audio-visual saliency prediction model for 360° video (Section 4). Design decisions are be supported by ablation studies (Section 5.4).
- Evaluation of the proposed model and comparison with the state of the art (Section 5.2 and 5.3).

This project was carried out in the Graphics and Imaging Lab research group, at the University of Zaragoza. The group conducts relevant research in the areas of physically-based rendering, image processing, computational imaging, virtual reality and applied perception, among others.

1.3 Timeline and tools

Figure 1.3 shows a Gantt chart of the schedule followed while working on this master's thesis. The time dedicated to this work was a total of 737 hours distributed over five months.

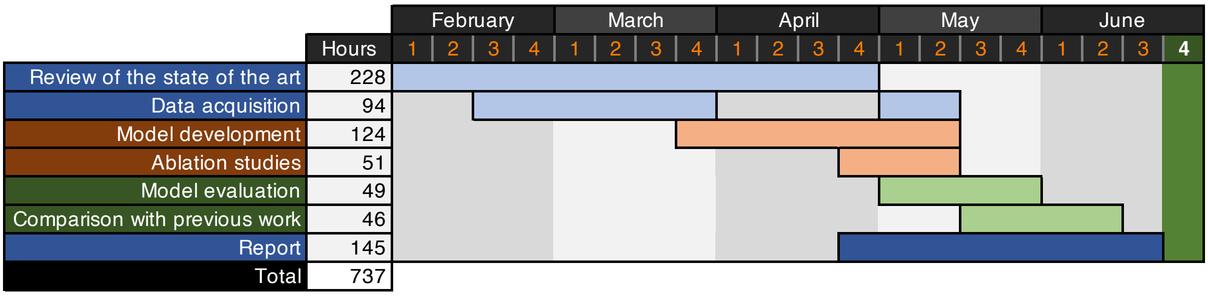

Figure 1.3: Gantt chart of the work schedule.

The saliency prediction model and programs developed in this work were implemented using the Python programming language and PyTorch, an open source machine learning framework. GitHub was used as a version control tool for the development of the programs.

2. Related work

This section provides a brief overview of previous works and how they relate to this project. The work is presented grouped into different categories:

2.1 Image saliency prediction

Visual saliency prediction attempts to model human visual attention. Classic attention models were implemented based on cognitive psychology using heuristic methods, often through handcrafted features. Itti et al. [4] proposed a solution that was the first successful attempt on visual attention modeling based on a neurobiological model and can be considered the most important work in this group. It was based on the bottom-up model, extracting low-level features of the images first to later combine them and obtain the final saliency map. The bottom-up approach was maintained by other works such as Itti et al. [5], which used improved handcrafted features; Bruce et al. [6], which was based on information maximization; or Harel et al. [7], which used a graph-based approach.

In contrast to these, other solutions were based on the top-down model, starting from higher level features instead of lower level ones. These solutions are often linked to specific tasks due to the complexity of developing a general solution for many different scenarios with this approach. Some works that follow this approach are Wolfe et al. [19], which proposed performing basic visual features over a large portion of the domain and complex operations over a limited portion; Gu et al. [20], which proposed a model based on the decision theory mechanism to predict regions of interest; or Judd et al. [21] combined low, middle and high-level image features through a linear SVM model.

Vig et al. [22] proposed one of the first data-driven solutions by using a convolutional neural network (CNN). This approach overcame the bottleneck caused by handcrafted features as it was able to automatically learn the most relevant features of an image, but was very limited due to the number of datasets and depth of the network. Thanks to the recent advances in the field of deep learning, many techniques flourished [23, 24, 25, 8, 9] that were able to learn and combine different level image features and obtain more accurate results. SalGAN [8] proposed using a generative adversarial network (GAN) and highlighted the importance of choosing an appropriate loss function, while works such as MSI-Net [9] used an encoder-decoder architecture to capture multiple scale features and obtain outstanding results. Like these, the model proposed in this work is based on CNNs, but it incorporates methods to adapt to the dynamic nature of video and additional challenges linked to 360° content.

While the mentioned works are able to obtain great results in saliency prediction for images, they only take into account static visual information and they do not include motion or auditory information, which are critical for saliency prediction in videos.

2.2 Traditional video saliency prediction

Compared to static image saliency prediction, the problem of visual attention modeling in videos has additional difficulties due to the added motion information and reduced observation times for each individual frame. Despite its difficulty, the demand for solutions in this field is very large and thus research has been continuously developing.

Early models based on handcrafted features that incorporated temporal information were initially proposed [26, 27] but their performance was significantly restricted by their inability to learn the most relevant features automatically.

Deep learning methods greatly improved this situation. Works such as Bak et al. [28] incorporated spatial and temporal features through a two-stream network. One part of the network processes spatial information and the other part processes temporal information. To obtain the final prediction, the two branches are combined through a fusion module. Similarly, the model proposed in this work uses a layer to combine features learnt by two separate branches. Leifman et al. [29] proposed merging the frame color data, optical flow map, and saliency map into a seven-layer CNN. By doing this, the model was able to obtain features combining data from the different sources without the presence of a bottleneck in the network such as a fusion module. The model proposed in this work also uses an input formed as a concatenation of data from different types of sources (visual and auditory). Gorji et al. [30] proposed the use of a CNN Long short-term memory (LSTM), which is a type of Recurrent Neural Network (RNN) that allows the model to learn temporal features across different frames. Unlike standard neural networks, LSTMs maintain a hidden state that gives them the ability to retain information for a long period of time.

The aforementioned works considerate visual and temporal information only and they do not include any information regarding auditory cues. However, the effect of auditory cues on visual attention has been studied [18, 17] and shown to have significant importance in some scenarios.

To incorporate auditory information, Song et al. [31] proposed a CNN model with two streams for visual and auditory features that, in certain cases, produced more accurate results than with visual information only. However, it only obtained better results when it was applied on speech sequences and it was provided with information on the source of the audio; when this was not the case, the model obtained even worse results than when working with visual information only. Sidaty et al. [32] proposed a technique that was an improvement over the previous solution by using a real-time audiovisual speaker localization method and thus, not needing to be provided with the source direction of the audio to improve its results. Lastly, Tavakoli et al. [33] proposed a model using a 3D Residual Network (ResNet) and decoder architecture that combined spatial and auditory features through suited for uncategorized video sequences.

The previous works incorporated spatio-temporal visual and auditory information, achieving excellent results for traditional video saliency prediction. However, this work focuses on saliency prediction for 360° content, which presents additional challenges for saliency prediction.

2.3 360° content saliency prediction

The development and popularization of VR has increased the demand and availability of 360° content, and with it, the need for models that represent and predict visual attention for this format. While models for traditional 2D images can be applied to 360° content, their performance is drastically decreased if no measures are taken to account for the additional challenges that this media poses.

Early works such as Fang et al. [34] proposed segmenting the image in superpixels and using the CIE Lab color space to extract handcrafted features before merging them into the final saliency map. However, heuristic-based methods were soon replaced by deep learning methods as these were shown to obtain more accurate results.

SalNet360 [35] proposed a model based on the previous SalNet [24] designed for 2D images. To palliate the distortions of 360° content, the model uses a cube-map representation of the scene where the six faces of the cube are treated as 2D planes. The obtained cube-map saliency map is converted into an equirectangular map, which is a more common representation. Similarly, SalGan360 [10] adapted SalGan [8] to 360° content, but this time by combining the results of the original model working on 360° images and the model adapted for cube-map projections. By using both cube-map and equirectangular representations, the proposed method was able to obtain more accurate results that with either one separately.

While the previous works are able to reduce the distortions in the 360° images by using cube-map representations or limiting the image to the viewport area, there are still distortions in the provided solutions and potential redundant computations. To solve this, other works like SphereNet [11] proposed replacing the classic kernel in CNNs with one that is adapted to work with equirectangular images using spherical coordinates: instead of combining directly adjacent pixels of an image, the adjacent positions are computed in spherical space and re-projected to the equirectangular projection. Other works such as Martin et al. [13] use the proposed spherical convolutions from SphereNet [11] and an encoder-decoder architecture inspired by U-Net [1] to predict saliency maps for single 360° images using equirectangular projections. Similarly, the model proposed in this work uses an U-Net like encoder-decoder module with spherical convolutions to account for distortions.

For 360° video, solutions were also heavily inspired by previous works targeted towards traditional 2D videos. Cheng et al. [12] proposed a CNN model that uses cube-map projections to palliate the distortion in the images combined with a convolutional LSTM to leverage the saliency maps of different frames. Zhang et al. [14] proposed a U-Net like [1] architecture using a new CNN with a spherical crown kernel and a Spherical Mean Square Error (SMSE) loss function that leverages the distortion in equirectangular images. This model obtained more accurate results than previous solutions but only used the saliency map of the previous frame and thus its temporal information was limited. Yanyu et al. [15] proposed a method that combined spherical convolutions from SphereNet [11], SMSE loss and LSTMs with a U-Net like [1] architecture; allowing the model to leverage distortions and use longer-term temporal data to generate its predictions.

Despite the importance of audio for 360° content saliency prediction being studied and documented [16], the aforementioned works do not incorporate any auditory information. Due to the complexity and novelty of the problem, very few works are found that study the usage of auditory cues in 360° video attention prediction. Chao et al. [3] proposed a model inspired by Tavakoli et al. [33] that combines cube-map and equirectangular projections to leverage the image distortions. To incorporate audio information, the model uses audio energy maps (AEMs), which represent the intensity of the audio coming from all different directions; and mel spectrograms, which represent the spectrum of frequencies of the audio signal. To the best of our knowledge, this is the only work that makes use audio information for saliency prediction in 360° video.

As proposed in the work by Chao et al. [3], the model proposed in this work uses AEMs to represent the spatiality of the audio and the mono audio wave spectrogram to characterize the audio. In contrast to the work by Chao et al., the proposed solution does not use any manually defined attention modelling mechanism. Instead, it automatically learns how to use the input data through deep learning. The main part of the proposed model consists of a U-Net like [1] architecture with spherical convolutions similar to the one proposed by Martin et al. [13] to leverage the distortions present in 360° content. The encoder is fed an input composed of concatenated layers: RGB images and AEMs of the different frames of the video are used, resulting in four layers for each frame. Alongside the encoder-decoder module, the model uses a ResNet module that extracts audio features from the audio wave spectrogram. The features learnt by the encoder and the ResNet are combined before being fed to the decoder to predict the final saliency map. To incorporate temporal features, 3D spherical convolutions are used, which are specially suited for short-term fast movements [36].

3. Theoretical foundations

With the purpose of making this document self-contained, this chapter explains some of the theoretical concepts that are used throughout the document. Only those concepts that are necessary to support this work are explained.

Section 3.1 provides an overview of attention modelling, saliency prediction and directional audio; as they are concepts that this work is based on. Additionally, Section 3.2 provides a summary of the deep learning techniques and architectures used in the proposed solution.

3.1 Attention modelling and saliency

In the field of computer vision, many efforts have been put towards the problem of modelling the mechanisms of human attention, specially focusing on visual saliency, which refers to the ability of objects or features to stand out and capture our attention. Despite significant progress, the problem of saliency prediction remains challenging, likely due to the fact that it is influenced by many different factors.

Current research on saliency prediction can be divided into two main different tasks, saliency or gaze prediction and salient object detection (SOD). Both tasks aim to model the likelihood of a specific area of an image or video to catch our attention but they do so in different abstraction levels. While saliency prediction uses information about human eye fixations and aims to predict the probability of the eyes of a person to stay on each pixel of an image or video, salient object detection tasks do this on an object level. Due to their differences, each of these tasks have their own application scenarios.

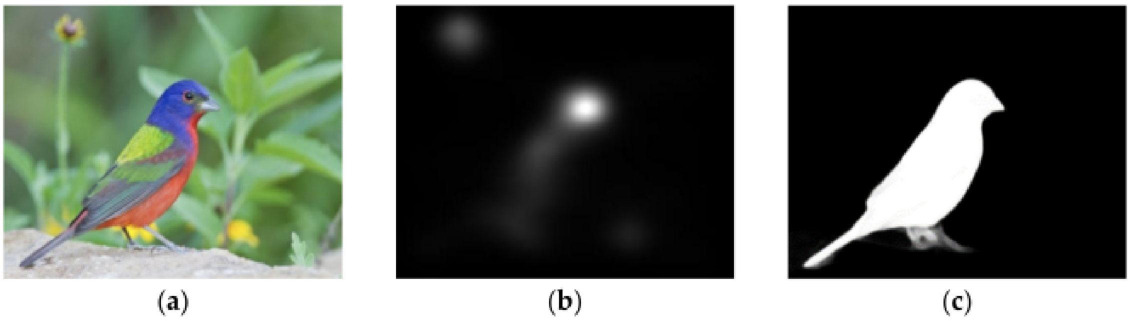

Figure 3.1: Two saliency detection tasks: (a) Original image, (b) Saliency prediction task, (c) Salient object detection task. Source: Yan et al. [37]

In this work, we focus on saliency prediction as it provides a more fine representation of human attention, delivering a saliency value for each individual pixel of an image, and it has a large number of applications such as image or video compression [38], visual SLAM (Simultaneous Localization and Mapping) [39], autonomous driving [40] or medical diagnosis [41].

3.1.1 Representing directional audio

Audio has an important effect on attention redirection[42], specially when it is directional as the user can easily locate the source of the signal. In 360° content, its effect could be more pronounced than in traditional media as the view is limited to a portion of the content and audio can help to redirect the gaze towards a location that is outside the current viewable area. For this reasons, directional audio information has been incorporated in this work, using the ambisonics format.

Ambisonics

Ambisonics [43] is a spherical surround sound format that provides the ability to encode sounds to a so-called B-format, maintaining the information about the directions of the sources. In contrast to other spatial sound formats, all sounds coming from different directions are treated equally, and it only requires four channels for full-sphere soundfields.

Despite the many advantages of ambisonics and its development dating back to the 1970s, it was not until recently that they became a commercial success as applications such as virtual reality showed their effectiveness in encoding a spherical soundfield which can be decoded matching the rotation of the user's head. In this work, ambisonics are used as a way to represent the spatial soundfield of the scenes and to compute a directional audio energy map (AEM) that is directly used to train the proposed model's network.

In the encoded B-format, each of the individual channels do not directly correspond to speaker feeds but contain components of the soundfield that are later combined to a speaker feed through a decoding process. This format has an order value that corresponds to the level of spatial detail that is utilized, related to the spherical harmonics. This way, a zero order ambisonic would correspond to a single mono channel and a first-order one to a slightly low resolution full-sphere soundfield. The first-order ambisonic, which is used in this work, encodes a soundfield into four individual channels:

$$\begin{array}{ll} W = S \cdot \frac{1}{\sqrt{2}} & \text{Omnidirectional information} \\ X = S \cdot \cos\phi\cos\theta & \text{Front-to-back information} \\ Y = S \cdot \sin\phi\cos\theta & \text{Left-to-right information} \\ Z = S \cdot \sin\theta & \text{Top-to-bottom information} \end{array} \tag{3.1}$$

where S is an audio signal, ϕ is the horizontal angle (azimuth), and θ is the vertical angle (elevation).

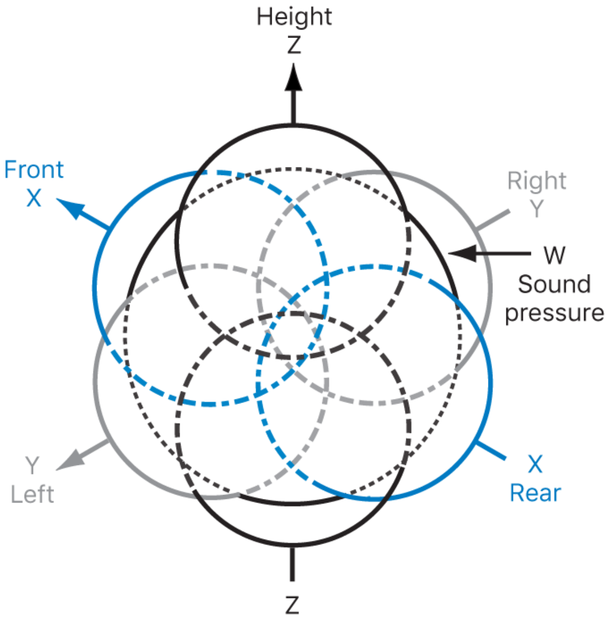

Figure 3.2: First-order ambisonics. Source: Apple [44]

Since the ambisonics format does not encode speaker feed directly, the encoded representation is independent from the number and distribution of speakers in the listener's setup, which provides an enormous flexibility compared to other formats. However, there is an additional necessary step that needs to be taken to convert the encoded channels to the feed signals for the playback speakers, known as decoding. One of the different decoding techniques that convert an ambisonics format to actual speaker signals is the projection decoding.

The projection decoding provides each speaker with a feed signal computed as the sum of the different channels weighted by the value of the corresponding spherical harmonic for the position of the speaker, defined as follows for the first-order ambisonics format:

$$p_i = \frac{1}{4}\left(W \cdot \frac{1}{\sqrt{2}} + X \cdot \cos\phi\cos\theta + Y \cdot \sin\phi\cos\theta + Z \cdot \sin\theta\right) \tag{3.2}$$

where p_i is the signal for the i-th speaker, (ϕ_i, θ_i) is the position of the i-th speaker, and W, X, Y, Z are the channels of the ambisonics B-format.

Audio Energy Map

Instead of directly using the ambisonics B-format directly to train the model proposed in this work, a more simple representation of the audio spatiality was used. Audio Energy Maps (AEMs), computed using the ambisonics B-format as proposed by Morgado et al. [45], provide a measurement of the intensity of the audio coming from different directions of the soundfield.

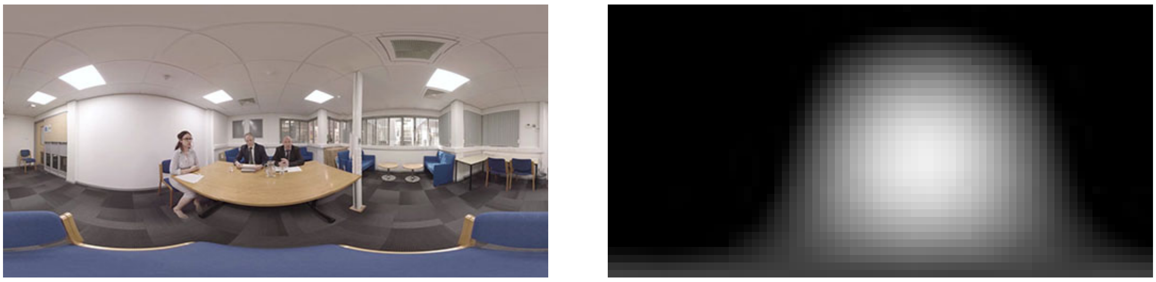

Figure 3.3: Example AEM. A frame of the 360° video (left) is shown along with the generated AEM (right). The AEM indicates the intensity of the audio coming from all different directions. Black values represent low audio intensity and white values high intensity.

Using the equirectangular projection (Section 3.2.1), an AEM is expressed in the form of a 2D grayscale image, where each pixel represent the intensity of the audio coming from its corresponding direction. To compute the AEM, a speaker feed signal is decoded for each pixel of the resulting image as described in Equation 3.2 and the root mean square (RMS) of the decoded audio signal is computed as described in Equation 3.3. Finally, the computed AEM is normalized so that the values are within the range $[0, 1]$.

$$RMS = \frac{1}{N}\sqrt{s_i^2} \tag{3.3}$$

where s_i is the i-th sample of the audio signal; and N is the total number of samples.

3.2 Deep Learning

In recent years, we have seen the rise of deep learning solutions for countless problems, in many cases, rendering most of the previously developed solutions obsolete in comparisons. This has been allowed by many different factors such as the increasing computational power provided by graphical processing units (GPUs), which has enabled to lower the long training times that these techniques require; the enormous amount of available data on the internet, resulting in large enough datasets to train the developed models; and the development of more advanced algorithms.

Machine learning techniques have shown particularly good results in tasks related to image or video such as recognition, segmentation, classification or enhancement. One of the key factors that has allows these techniques to obtain such good results in such tasks is the use of convolutional neural networks (CNNs). This type of networks work by applying a convolution kernel that is slid along the input to learn translation-invariant characteristics of the input known as features.

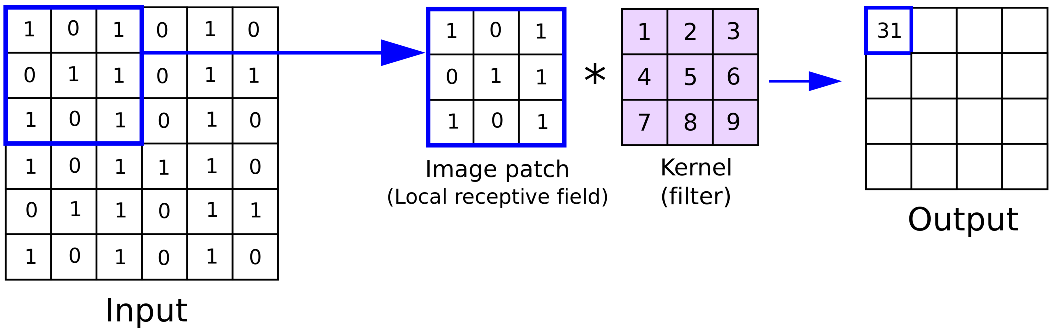

Figure 3.4: Traditional convolution. Source: Anh H. Reynolds [46]

CNNs provide an effective method to obtain features regarding the nearby pixels on the input and, by applying them on multiple levels, to learn more complex features and cover the entire input area.

3.2.1 Equirectangular images

One advantage of 360° content over 3D content is that it can be encoded using classic 2D media formats and compression techniques by making use of a spherical projection, which pairs each position on the surface of the sphere to a position on the surface of the plane. However, it comes with some caveats as neither areas nor distances are preserved. As such, measures have to be taken into account to overcome the distortion introduced by the projection depending on each application.

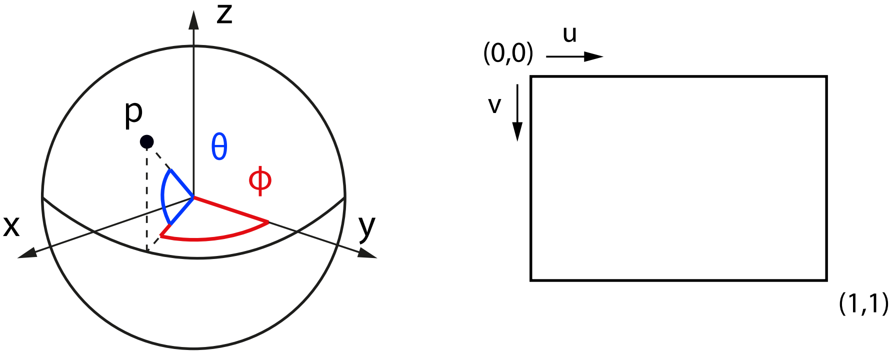

Figure 3.5: Spherical projection.

The equirectangular projection is one of the most common projections used nowadays. It maps the longitude of the sphere directly to the horizontal coordinate, and latitude directly to the vertical coordinate. The conversion from spherical coordinates to plane coordinates and vice-versa can be defined as follows:

$$(u, v) = \left(\frac{\phi + \pi}{2\pi}, \frac{\theta - \frac{\pi}{2}}{\pi} \right) \tag{3.4}$$

$$(\phi, \theta) = \left(\left(u - \frac{1}{2} \right) 2\pi, \left(v + \frac{1}{2} \right) \pi \right) \tag{3.5}$$

where (u, v) are the coordinates on the plane, with values ranging from $(0, 0)$ (top-left) to $(1, 1)$ (bottom-right); and (ϕ, θ) are the coordinates on the sphere, with values ranging from $(0, \frac{-\pi}{2})$ to $\left(2\pi, \frac{\pi}{2}\right)$.

Like other spherical projections, this method has some downsides. It produces a very distinct curved look to the images and a sub-optimal pixel density distribution as the equator, which is normally the most important part, receives the lowest pixel density in the projection plane.

Despite its drawbacks, it is one of the most used spherical projections for 360° content due to its simplicity, which is one of the reasons that it is used in this work. Compared to other techniques, there is a very broad set of data available that uses the equirectangular projection and, while the amount of 360° content available is still very limited compared to other traditional

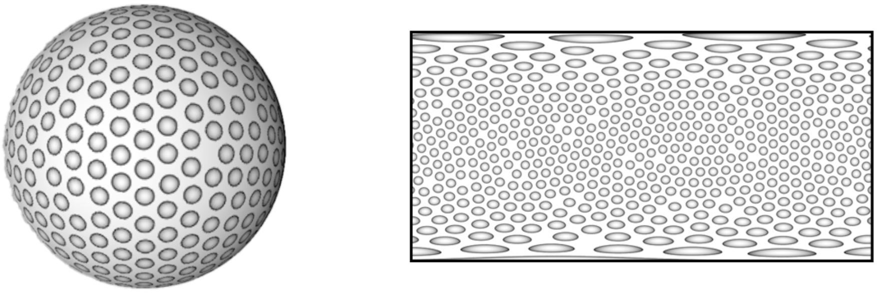

Figure 3.6: Equirectangular projection of sphere (left) onto a plane (right).

forms of media, using a commonly used projection provides allows the use of a sufficiently large dataset, which is specially important when working with deep learning techniques.

3.2.2 Spherical Convolutions

Regular CNNs work extremely well on traditional image and videos as the weights learnt on one patch of the image or video frame can be translated to any other patch. However, when applied to images that utilize spherical projections such as the equirectangular one, the results are much inferior due to the distortions inherent in the projections. The weights learnt for a classic CNN kernel on a specific position of the input image cannot be correctly extrapolated to a different position as the distortions on the two patches are different.

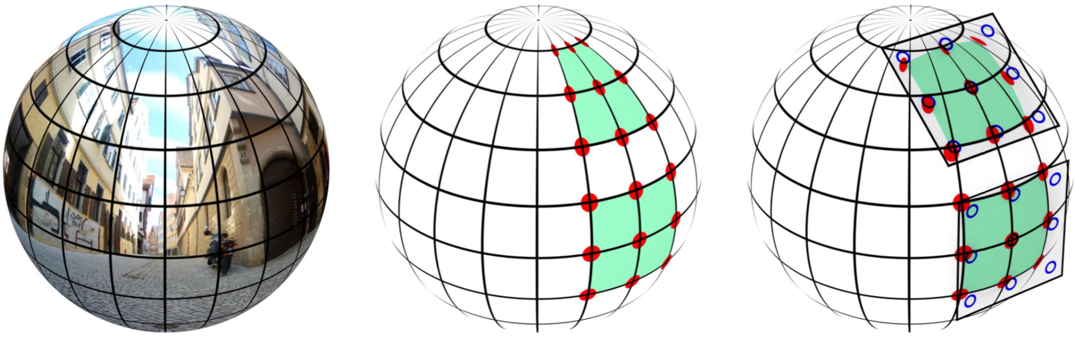

Figure 3.7: 360° image (left), regular convolution (middle), spherical convolution (right). Source: SphereNet [11]

To overcome this difficulties, recent works propose the use of different convolutions that account for the distortion inherent to the projection. This way, a CNN can be applied on the spherical input with a patch that adapts to the distortion. The model proposed in this work makes use of the spherical convolutions presented in SphereNet [11], which are an adaptation of traditional convolutions that takes into account the distortion in images that use the equirectangular projection. This convolutions don't apply the kernel directly to adjacent pixels, instead, the kernel is applied for the neighbor pixels in the spherical domain. To do this, the kernel is sampled as the projection on the sphere of a patch tangent to its surface (Figure 3.7).

As it can be seen in Figure 3.7, regular convolutions in equirectangular images suffer from a very large distortion in areas close to the poles. Using the spherical convolutions presented in SphereNet [11] results in a sampled patch shape that is invariant to both longitudinal and latitudinal rotations.

3.2.3 Used architectures

U-Net

In this work, a U-Net [1] like architecture is used to process the input color frames and AEMs. U-Net like architectures have been shown to obtain outstanding results in both image [47] and video [48] tasks as they are able to obtain very high localization accuracy while capturing well the context.

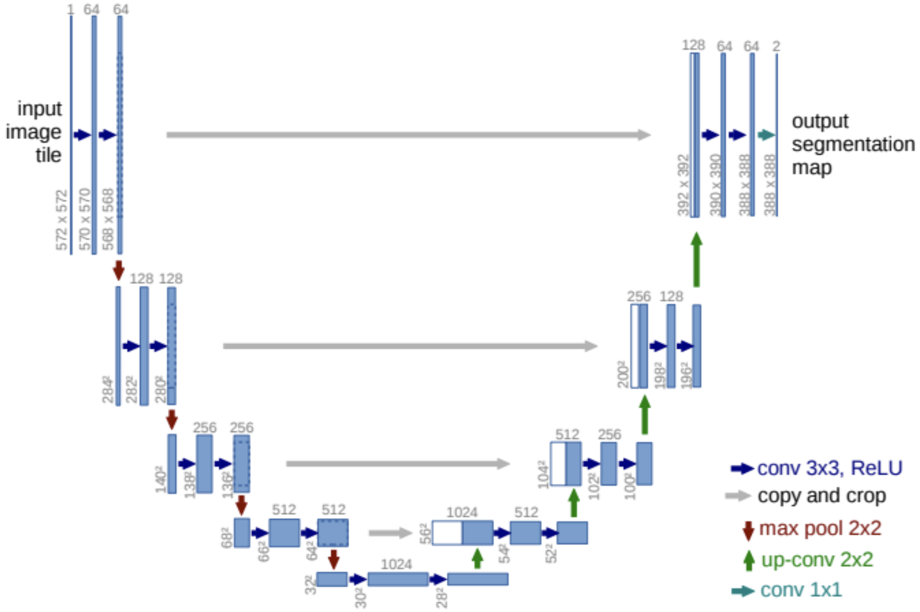

Figure 3.8: U-net architecture. Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations. Source: U-Net [1]

One of the key ideas of U-Net is the incorporation of connections between the encoder and decoder modules along the different depths of the architecture (Figure 3.8). These connections allow the architecture to obtain a high localization accuracy that provides a notable advantage over more traditional encoder-decoder architectures while maintaining a good use of the context.

ResNet

The model proposed in this work makes use of ResNet [2] architecture with 18 layers to learn the characteristics of the audio signal of the input clip.

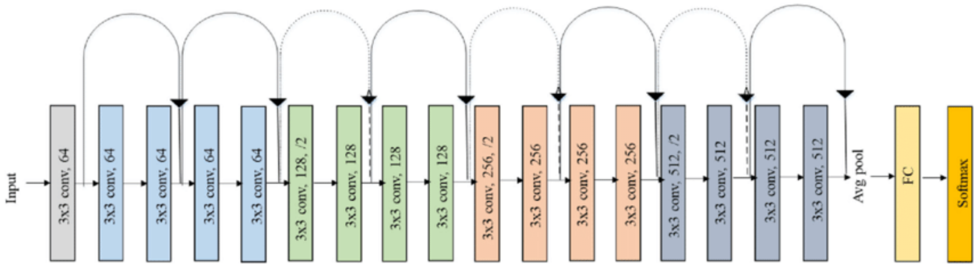

Figure 3.9: ResNet with 18 layers. Source: ResNet [2]

ResNets are networks that have a very high efficiency thanks to the incorporation of residual blocks that are able to maintain a low error rate deep in the network. Essentially, residual blocks are composed by two layers that use a skip connection or shortcut between the input of the first layer and the output of the second one; these blocks can be stacked on top of each other to create networks of different depths.

4. A model for audio-visual saliency prediction in 360° video

This section describes the proposed model for audio-visual saliency prediction in 360° video. The model, based on deep learning, combines a ResNet network and a U-Net like encoder-decoder architecture with 3D Spherical Convolutions. It receives as input RGB and AEM frames, as well as the spectrogram of the audio wave. The network combines these to learn spatiotemporal auditory and visual features and predict the final saliency map.

Section 4.1 provides a detailed description of the architecture and the different parts that constitute it. Next, Section 4.2 explains the type of spherical convolutions that the model is based on. In addition to that, Section 4.3 describes the dataset that was used to train the model. Finally, Section 4.4 summarizes the details of the training process.

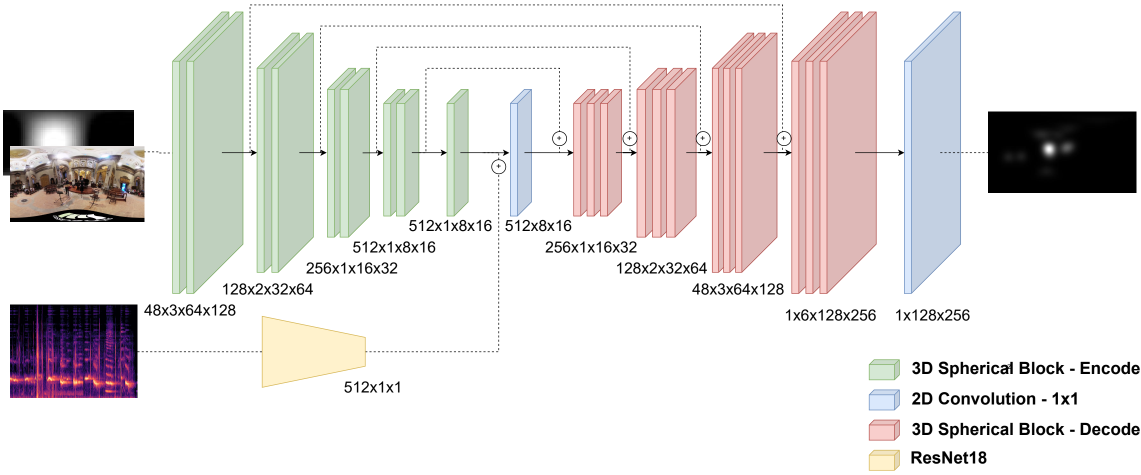

Figure 4.1: Architecture of the proposed model. The model is formed by a U-Net like encoder-decoder module and a ResNet encoder. The encoder-decoder branch, which uses 3D Spherical Convolutions, is fed a concatenation of RGB values and AEMs. The ResNet encoder takes as an input a mel spectrogram of the audio signal. A 1x1 2D convolution layer is used to combine the features learnt by both encoders before feeding them to the decoder module to obtain the final saliency prediction.

4.1 Architecture

The proposed architecture consists of two main branches:

Encoder-Decoder. It is a U-Net like encoder-decoder module that utilises Spherical Convolutions [11] to palliate the distortions in the equirectangular projection. To leverage temporal information, it uses a new type of 3D Spherical Convolutions that has been implemented in this work. These layers are grouped in 3D Spherical Blocks, which are described in Section 4.1.1. The module incorporates both spatio-temporal visual information and spatio-temporal information about the sources of the audio, by being fed a concatenation of AEMs and RGB values. Using this input allows the model to automatically combine the visual and spatial audio data without the need for a manually defined attention modelling technique, as opposed to other works [3]. To the best of our knowledge, this is the first work for saliency prediction in 360° content that is able to do so.

ResNet18. A ResNet architecture with 18 layers is used to learn features about the audio signal. This architecture was chosen based on the fact that it has been shown to be very effective due to the inclusion of skip connections (see Section 3.2.3). The output of the ResNet encoder is connected to the encoder-decoder module through a combiner layer with the purpose of incorporating audio features into the final prediction.

Figure 4.1 shows a diagram of the complete architecture of the model. The main part of the architecture is composed of the encoder-decoder module, which is fed AEM and RGB frames. The only auditory information fed to the encoder module is through AEMs, which represent sources of the audio. To learn features about the characteristics of the audio signal, a ResNet architecture is used that is fed with a mel spectrogram of the audio signal. A 1×1 2D convolution layer serves to combine the features learnt by the spherical encoder module and the ResNet module. The final saliency prediction uses features learnt from all input sources.

4.1.1 Detailed description

Different parts that constitute the proposed model are described in detail below.

3D Spherical Block. Each block consists of a 3D Spherical Convolution, which learns spatio-temporal features, and accounts for the distortion in the equirectangular projection; a batch normalization layer, used to make the model faster and more stable; and a ReLU layer.

3D Spherical Blocks are used on different levels of the encoder and decoder modules.

Encoder. It takes a set of T consecutive RGB and AEM frames as an input and outputs a 512 channel feature matrix by down-sampling the input image, resulting in a shape of $512 \times (W/16) \times (H/16)$ where W and H are the width and height of the input video. The encoder is composed of 4 levels, each consisting of two 3D Spherical Blocks that handle the distortion in the panoramic images and incorporate temporal information. The resulting feature vector contains both spatiotemporal visual and auditory information. However, the learnt auditory features are limited to the intensity of the audio coming from all directions.

ResNet18. It takes the spectrogram of the audio wave from the previous 0.5 seconds of the frame which saliency is to be predicted as input. The output is a 512 channel feature vector. This allows the model to learn features regarding the characteristics of the audio.

Combiner. The result of the ResNet module is expanded to a $512 \times (W/16) \times (H/16)$ shape and concatenated with the result of the encoder module along the first dimension, resulting in a $1024 \times (W/16) \times (H/16)$ shape. This is passed through a 1x1 convolution network to obtain a $512 \times (W/16) \times (H/16)$ auditory and visual spatiotemporal feature matrix. By doing this, all learnt features (visual features, features regarding directions of the audio and features regarding characteristics of the audio) are combined together.

Decoder. It takes the combined feature matrix as input and predicts the resulting panoramic saliency map through a series of 3D Spherical Convolutions. Similarly to the encoder module, it is composed of 4 different levels, but each level consists of three 3D Spherical Blocks. As proposed by U-Net[1], the layers of the decoder concatenate the intermediate representations from the down-sampling path in the encoder module to improve the model's efficiency. The resulting $T \times W \times H$ shape is converted to a $1 \times W \times H$ saliency map through a final 1x1 convolution layer at the end of the network.

4.2 3D Spherical Convolutions

To adapt to the dynamic nature of video and incorporate information regarding motion, the proposed model makes use of 3D convolutions. Instead of using a 2D kernel, this type of convolutions use a 3D kernel that is slid across three directions. This allows to apply the convolutions on volumetric data, producing an output that is three-dimensional as well. In the case of video, the additional dimension is mapped to the different frames of the video, which gives the model the ability to incorporate short-term temporal information into the final solution.

To mitigate the distortion inherent in the equirectangular projection, a new 3D Spherical Convolution has been implemented in this work that implements the sampling technique proposed in Spherical Convolutions (Section 3.2.2) for the two dimensions of the image space. Thus, one dimension of the 3D kernel is mapped directly to the different frames of the video, and the other two dimensions are sampled as the projection of a patch tangent to the surface of the sphere. This effectively combines the best of both techniques, successfully palliating image distortions and providing the ability to learn spatio-temporal features.

4.3 Dataset

To train the proposed model, the ASOD60k dataset [49] was used, which consists of a very diverse set of videos with different real-world scenes (e.g. indoors/outdoors), scenarios (e.g. sports, concerts, interviews...), motion patterns (e.g. static/moving camera), and types of objects present (e.g. humans, animals, instruments). The dataset is composed of 67 4K (3840 x 2160) videos with frame rates varying from 24fps to 60fps, with a duration of 29.6s on average.

The ground truth gaze data of 40 different participants is provided, which was recorded using a HTC Vive HMD embedded with a Tobii eye tracker. For all frames, the recorded position of the eyes on the 360° image is given for the different participants.

The dataset is split into three different groups: 40 videos (60%) as the train set, which will be used to train the model; 13 videos (20%) as the validation set, which will be used to obtain an unbiased evaluation of the model fit on the dataset while tuning the hyper-parameters; and 14 videos (20%) as the test set, which will be used to obtain the final evaluation of the model fit on the dataset.

The process followed to generate the saliency and audio data taken as input by the model is described in the following sections.

4.3.1 Saliency data

In order to train the model, saliency maps have been computed from the provided gaze data [42]. First, a fixation map is constructed for each frame of a video by counting the number of fixations per pixel. Then, the whole map is divided by the maximum value across all pixels. Finally, this fixation map is transformed into a saliency map by applying a Gaussian filter with standard deviation of 4 degrees. The Gaussian filter's horizontal radius is scaled to account for the distortion present in equirectangular images by increasing it towards the vertical limits of the image. Figure 4.2 shows a sample saliency map computed from a fixation map.

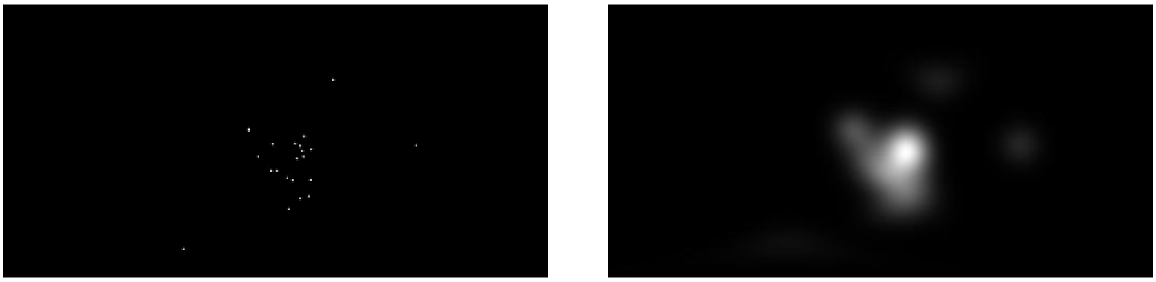

Figure 4.2: Example of the conversion process of a fixation map (left) to a saliency map (right). A Gaussian filter with a horizontal radius that scales to account for the distortion introduced by the equirectangular projection is applied to the fixation map.

To reduce memory needs and computation times, the provided frames of the videos have been scaled down to a final resolution of 256 × 128. In the first second of the videos, all participants looked at the center of the image while they waited for the video to begin. To compensate for this, the first 2 seconds of all videos are trimmed before training. On top of that, the length of the videos has been trimmed to a total duration of 600 frames (20 seconds on average).

4.3.2 Audio data

Audio energy maps (AEM) are calculated using the four channels (W, X, Y, Z) in ambisonics (see Section3.1.1) as proposed by Morgado et al. [45]. For each frame, the previous 1/6 seconds of audio in the ambisonic signals are used to generate an energy map with a resolution of 36×72 pixels.

Additionally to AEMs, for each frame a mel spectrogram (Figure 4.3) is generated as proposed in DAVE [33]. This spectrogram is a representation of the spectrum of frequencies of the audio signals that uses the mel scale for the frequency dimension. This scale accounts for human perception of sound volume. For the different frames of the video, the mono audio signal computed from the ambisonics is resampled in 16KHz and converted into mel spectrograms.

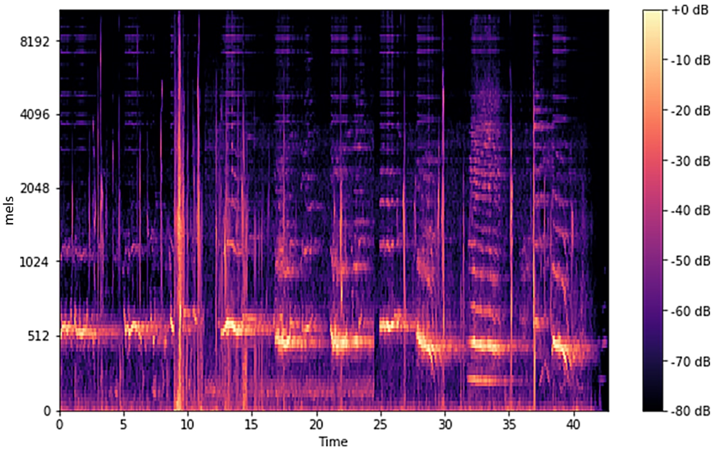

Figure 4.3: Example mel spectrogram of an audio signal. The intensity of the different frequencies of the audio signal across the duration of the signal is expressed in decibels (dB). The x-axis represents the time dimension (seconds), and the y-axis represents the frequency dimension (mels) using the mel scale.

4.3.3 Data augmentation

As the size of the dataset is still quite limited, a series of data augmentation techniques have been applied to increase the volume of data used to train the model operating on the already available videos.

Typically, data augmentation techniques generate variations of the existing images or videos by cropping them or applying different kinds of color alterations, such as adding grain or altering the brightness. Due to the difference in nature between traditional media and 360° content, a different method has been used in this work to augment the dataset.

360° images represent pictures that wrap around a whole sphere. During the process of encoding the images, a decision has to be made regarding the position of the horizontal limits of the image on the sphere. This way, the left and right limits could be located on any longitudinal position. As a result, the same spherical video could be encoded with different results.

In this work, the dataset has been augmented by applying a set of longitudinal shifts to the spherical images. Each of the videos have been augmented to three new variations by applying shifts to the RGB and AEM frames of 90°, 180° and 270° angles. This way, the final size of the dataset is effectively multiplied by four (see Figure 4.4).

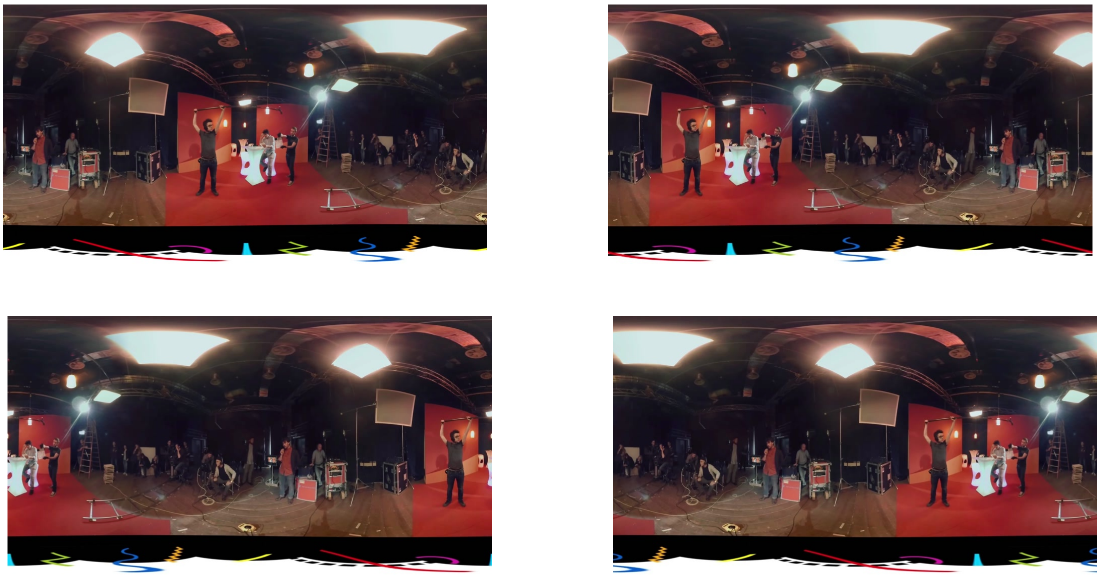

Figure 4.4: Example frame of a video augmented using longitudinal shifts. The original frame (top-left) is shown along with longitudinal shifts of 90° (top-right), 180° (bottom-left) and 270° (bottom-right).

4.4 Training

To train the model, the videos in the dataset were split in sequences of 6 consecutive frames (100 sequences per video). The model was trained with the different sequences using an Adam optimiser with a batch size of 2 sequences and a learning rate equal to 10^{-4}.

The training process took around 42 hours for a total count of 12 epochs for the final model selected. It was performed on a MSI Stealth GS66 laptop equipped with an Intel Core i7-12700H CPU and a Nvidia Geforce RTX 3090 GPU (24GB) connected through a Thunderbolt 3 eGPU. The programs developed in this work were implemented using the Python 3.7 programming language and the PyTorch 1.10.2 machine learning framework.

4.4.1 Loss function

Binary Cross Entropy (BCE) was used as the loss function to train the model, a widely-used loss function in works regarding saliency prediction. It measures the overall disparity between the ground truth and the prediction as dense probability maps and is defined as follows:

$$BCE(G,P) = -\frac{1}{N}\sum_{i,j} G_{i,j} \cdot \log\left(P_{i,j}\right) + \left(1 - G_{i,j}\right) \cdot \log\left(1 - P_{i,j}\right) \tag{4.1}$$

where $G_{i,j}$, $P_{i,j}$ are the ground truth and predicted saliency values at pixel (i,j) respectively and N the number of pixels.

5. Results and evaluation

This section provides a performance analysis of the proposed model for 360° video saliency prediction and a comparison with a state-of-the-art work. Section 5.2 shows a subset of results obtained with the proposed model and Section 5.3 presents a quantitative and qualitative comparison with AVS360 [3], a state-of-the-art work for 360° video saliency prediction that uses visual and auditory information. Additionally, different studies have been performed to support the choices made during the design of the proposed solution: Section 5.2.1 presents a comparison between the proposed model and a smaller-sized variant of it, and Section 5.4 exhibits a series of ablation studies that analyze how the different parts of the model affect its final performance.

5.1 Evaluation metrics

To evaluate the performance of the model, different metrics that appear frequently in saliency prediction literature have been used. In this section we briefly introduce them, please refer to Appendix A for the complete definitions.

Correlation Coefficient (CC): it is a statistical function that measures the dependence or linear correlation between the predicted and ground truth saliency maps.

Normalized Scanpath Saliency (NSS) [50]: it takes a saliency map and a set of fixations as input and measures the values of saliency at the fixation locations.

Kullback-Leibler Divergence (KLD): it is a statistical function that measures the overall dissimilarity between the ground truth and predicted saliency maps.

Similarity Metric (SIM): it is a function that measures the amount of overlap (and thus overall similarity) between the predicted and ground truth saliency maps.

Bhattacharyya coefficient (BC): it is a function that, as the SIM metric, measures the overall similarity between the predicted and ground truth saliency maps.

To account for the distortion inherent in the equirectangular projection, the aforementioned metrics are weighted as proposed by Gutiérrez et al. [51]; the values are multiplied by sine of the latitude, increasing the importance of positions near the equator on the final value.

5.2 Proposed method results

As mentioned in Section 4.3, the ASOD60k [49] dataset has been divided into three different groups: train set (40 videos), validation set (13 videos) and test set (14 videos). This provides a method to obtain an unbiased evaluation of the model fit while tuning the hyper-parameters using the validation set, and a method of obtaining the final unbiased evaluation of the model fit on the dataset using the test set.

Figure 5.1 shows sample results obtained with the proposed model using the test set from the ASOD60k [49] dataset and Table 5.2 shows the quantitative evaluation of the proposed model using the metrics previously described in Section 5.1. For a pair of test videos, four RGB frames and the respective AEMs are shown along with the ground truth saliency maps and the predictions computed using the proposed model. As it can be seen, the proposed model is able to obtain saliency predictions that are very close to the ground truth, being able to produce fine-detailed results that focus correctly on the parts of the videos that are more salient. In the sequence labeled "Train" shown in Figure 5.1, it can be observed that the model is able to react correctly to the audio energy maps, making the salient parts of the frame to shift to the right when the train is crossing.

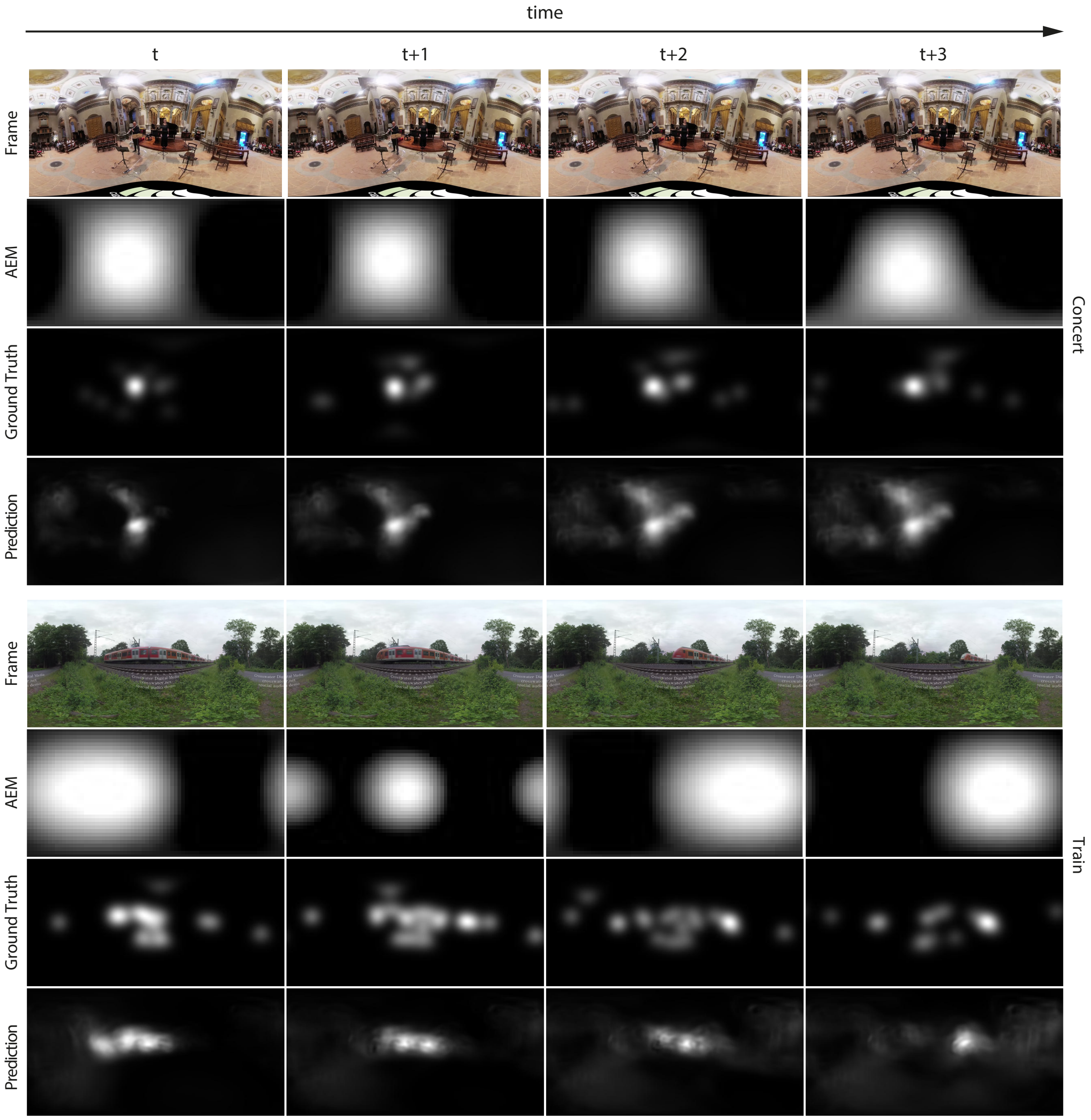

Figure 5.1: Results obtained using the proposed model for a series of test sequences. The horizontal axis represents time. The vertical axis shows from top to bottom: The RGB frame, AEM, the ground truth saliency and the proposed model's saliency prediction. Black values represent zero probability of being observed and white values high probability. Note that the proposed model is able to overall predict the salient areas correctly, producing a fine detailed result.

5.2.1 Model scalability

During the design phase of the proposed model, decisions about the number of layers and sizes of the feature vectors had to be made taking into consideration the effect that they would have on the overall performance and the available memory in the GPU used for training. This way, inspired by the original implementation of U-Net [1], the final design fits within the available memory and consists of four downsampling/upsampling blocks on either side of the encoder-decoder module, with the size of the feature vectors increasing the deeper the layer is up to 512 features on the deepest level.

As the model uses nearly all the available memory during the training phase, to analyze the effect of the size of the model on its final performance, a scalability study has been performed that compares the accuracy of the results obtained with the proposed model and a variant that uses a smaller size for the feature vectors. The implemented variation of the model uses a total of 256 features instead of 512 at the last layer of the encoder module. Table 5.1 shows the quantitative evaluation of the proposed model and the smaller size variation. The results show that the smaller variation obtains worse results than the proposed model, which indicates that choosing a higher number of features was beneficial for the performance of the model.

Due to the aforementioned hardware limitations, it was not possible to perform a scalability study of the proposed model with larger sizes. Future work that overcomes this limitations could be done, analyzing the performance of larger-sized variations of the model. This is further discussed in Section 6.1.

	CC ↑	NSS ↑	SIM ↑	BC ↑	KLD ↓
512 features	**0.457** **(0.178)**	**1.987** **(0.934)**	**0.385** **(0.095)**	**0.628** **(0.102)**	**10.268** **(4.079)**
256 features	0.325 (0.118)	1.374 (0.864)	0.314 (0.094)	0.547 (0.114)	12.609 (4.319)

Table 5.1: Quantitative comparison of the proposed model and a smaller size variation. Arrows indicate whether a higher or lower values are better for each metric. Bold letter highlights the best result for each metric. The values show the mean score among the different videos in the dataset and the standard deviation in brackets. The individual scores for each video can be found in the Appendix B.

5.3 Comparison with the state of the art

This sections presents a comparison of the proposed model and the state-of-the-art work, AVS360 [3]. This work is, to the best of our knowledge, the only work for 360 video saliency prediction that utilises visual and spatial auditory information. Instead of using spherical convolutions, to overcome the distortions caused by spherical projections, this work handles the video frames through two branches: one handles the frames using the equirectangular projection, which are processed using traditional convolutions; and the other one uses the Cube Padding technique, which renders the 360° images onto a cube and processes the six faces of the cube while utilizing the connectivity between them. It incorporates spatial auditory information through a hand-crafted attention modeling technique that combines AEMs and a bias that makes the pixels closer to the center of the equirectangular image more salient as this is often the case [42]. In contrast to this, the model proposed in this work automatically learns how to utilise AEM to obtain the final saliency prediction, and it does not incorporate any bias towards the center of the image.

Figure 5.2 shows some results obtained with both models on a small set of videos from the test set that were not longitudinally shifted. While AVS360 [3] tends to generate very large and blurry salient areas, the proposed model is able to generate saliency predictions that are overall closer to the ground truth and more fine detailed. In contrast to AVS360, the proposed model is able to focus on multiple different objects present in the scene, predicting smaller and disconnected salient areas.

Figure 5.3 shows some results predicted by both models on a pair of videos from the test set that were augmented with a longitudinal shift of 180°. The proposed model is able to generate accurate results even if the salient parts of the video are far from the center of the image, while the results predicted by AVS360 [3] tend to generate high saliency values near the center. This is a result of the attention modelling technique used in AVS360, that combines the AEM with a bias towards the center of the frame to encourage high saliency values in those areas. Although it is often the case that salient features are located near the sources of the audio or the center of the image, this is not always true and the data augmentation that was performed showcases the limitations of this techique clearly. In contrast to this method, the proposed model learns how to use AEMs automatically without incorporating any bias towards the center of the image. The results show that it is able to do so successfully, improving its predictions and overcoming the limitations that the method used in AVS360 presented.

Table 5.2 shows the quantitative evaluation of the different models using the metrics previously described in Section 5.1. As it can be seen in the table, the proposed model obtains better results than AVS360 for all metrics.

	CC ↑	NSS ↑	SIM ↑	BC ↑	KLD ↓
Proposed Model	**0.402 (0.202)**	**1.731 (0.975)**	**0.360 (0.106)**	**0.598 (0.120)**	**11.368 (4.497)**
AVS360	0.376 (0.167)	1.480 (0.727)	0.347 (0.095)	0.584 (0.114)	12.356 (4.024)

Table 5.2: Quantitative comparison of the proposed model and AVS360[3]. Arrows indicate whether a higher or lower values are better for each metric. Bold letter highlights the best result for each metric. The values show the mean score among the different videos in the dataset and the standard deviation in brackets.

5.4 Ablation studies

To support the effectiveness of the different decisions taken during the design of the proposed model's architecture, different ablation studies have been performed. Due to time constrains, the different variants of the model compared in this section were trained without the data augmentation mentioned in Section 4.3.3. The results are shown in Table 5.3.

AEM	Audio Spec.	3D Conv.	Spher. Conv.	CC ↑	NSS ↑	SIM ↑	BC ↑	KLD ↓
✓	✓	✓	✓	0.457 (0.178)	1.987 (0.934)	0.385 (0.095)	0.628 (0.102)	**10.268 (4.079)**
	✓	✓	✓	0.383 (0.176)	1.621 (0.817)	0.336 (0.090)	0.575 (0.105)	12.351 (3.848)
✓		✓	✓	0.386 (0.213)	1.614 (0.945)	0.343 (0.105)	0.578 (0.116)	12.112 (3.915)
✓	✓		✓	0.379 (0.176)	1.619 (0.857)	0.337 (0.094)	0.573 (0.108)	12.262 (4.220)
✓	✓	✓		**0.458 (0.200)**	**2.011 (1.063)**	**0.393 (0.113)**	**0.631 (0.119)**	10.494 (4.396)

Table 5.3: Ablation Studies. Quantitative comparison of the proposed model and the different variations implemented. Check marks indicate the elements included in each model. Arrows indicate whether a higher or lower values are better for each metric. The values show the mean score among the different videos in the dataset and the standard deviation in brackets. The individual scores for each video can be found in the Appendix B. Bold letter highlights the best result for each metric. Note that the best results, obtained with traditional convolutions, are marginally larger than the ones obtained using the proposed model and have higher variance. This is further discussed in Section 5.4.4.

In the following sections we discuss the results and contributions of each component.

5.4.1 Audio Energy Maps

Feeding AEMs as an input to proposed model provides a way for it to know what parts of the video does the recorded audio come from and potentially, learn on what parts of the video to focus combining this information with the visual information. To study the effect that this inclusion has on the overall performance of the model, a variation of the model where the directional information of the audio is not provided has been trained. Thus, only RGB data of each frame and audio spectrograms are fed to the model. The results can be observed in the second row of Table 5.3.

The performance of the model decreases consistently for all the metrics used when it is deprived from AEM data. This shows that using AEMs is an effective method to provide the model with the ability to locate the sources of the audio and improve its performance.

5.4.2 Audio spectrograms

Audio spectrograms give the ability to the model to learn characteristics about the audio signal and combine this information with visual one to know what parts of the video to focus on (e.g., the model would tend to focus on faces if the audio signal is detected to be a conversation). Song et al. [31] proposed a method that incorporated auditory information but the proposed solution's performance depended very much on the information about the audio source being available. For this reason, in this ablation study, a variation of the proposed model that is deprived of spectrograms and one that is deprived of both audio spectrograms and AEMs have been trained. The results can be observed in the third row of Table 5.3.

Results show that, although the model is still provided with AEMs, the performance decreases for all metrics when it is deprived of mel spectrograms. This suggests that the model is not able to utilize information about the sources of the audio as efficiently when the characteristics of the audio are not known.

5.4.3 3D convolutions

In the proposed solution, 3D (spherical) convolutions are used so that the proposed model can utilise temporal information to obtain the final saliency prediction for each frame. To assess the effect that including this temporal information has on the accuracy of the model's results, in this ablation study the 3D spherical convolutions in the encoder and decoder modules are substituted with 2D spherical convolutions. In the same way, traditional 3D convolutions are substituted with 2D convolutions in the audio branch. As a result, instead of feeding the model with a set of consecutive frames, only one frame is fed at a time for the solution to predict its saliency. The results can be observed in the fourth row of Table 5.3.

As it can be seen in the results, when the model does not utilize temporal data, the performance of video saliency prediction is substantially reduced. This shows that motion has an important effect on visual saliency and including this information could be crucial to obtain good predictions.

5.4.4 Spherical convolutions

To take into account the distortion present in equirectangular images, Spherical Convolutions are used in the proposed model. In this ablation study, Spherical Convolutions are substituted with traditional convolutions in the encoder and decoder modules to evaluate their impact in the model's overall performance. The quantitative results can be observed in the fifth row of Table 5.3. Additionally, Figure 5.4 shows sample results obtained for a test sequence with the proposed model and the variation implemented in this section.

The quantitative results obtained for most metrics when substituting spherical convolutions with traditional ones are marginally better, being too close to be conclusive. The results can be partially explained by the weight technique used for the metrics, which makes values closer to the poles have less impact on the final result. This is done to account for distortions in the equirectangular projection, as items of the same size would appear larger when located towards the poles than when located near the equator of the image. Traditional convolutions do not account for this distortion and, as such, the results that they produce would be the worst in areas near the poles, where the distortion is the largest. This behavior is not shown by the quantitative results (Table 5.3), as the weighting technique greatly reduces the effect that this specific cases have on the final result. However, it can be clearly seen in the qualitative results (Figure 5.4). While both models obtain similar results in areas near the equator of the image, it can be observed that the proposed model obtains results closer to the ground truth on areas near the poles. This shows the effectiveness of using spherical convolutions instead of traditional ones when working with equirectangular images.

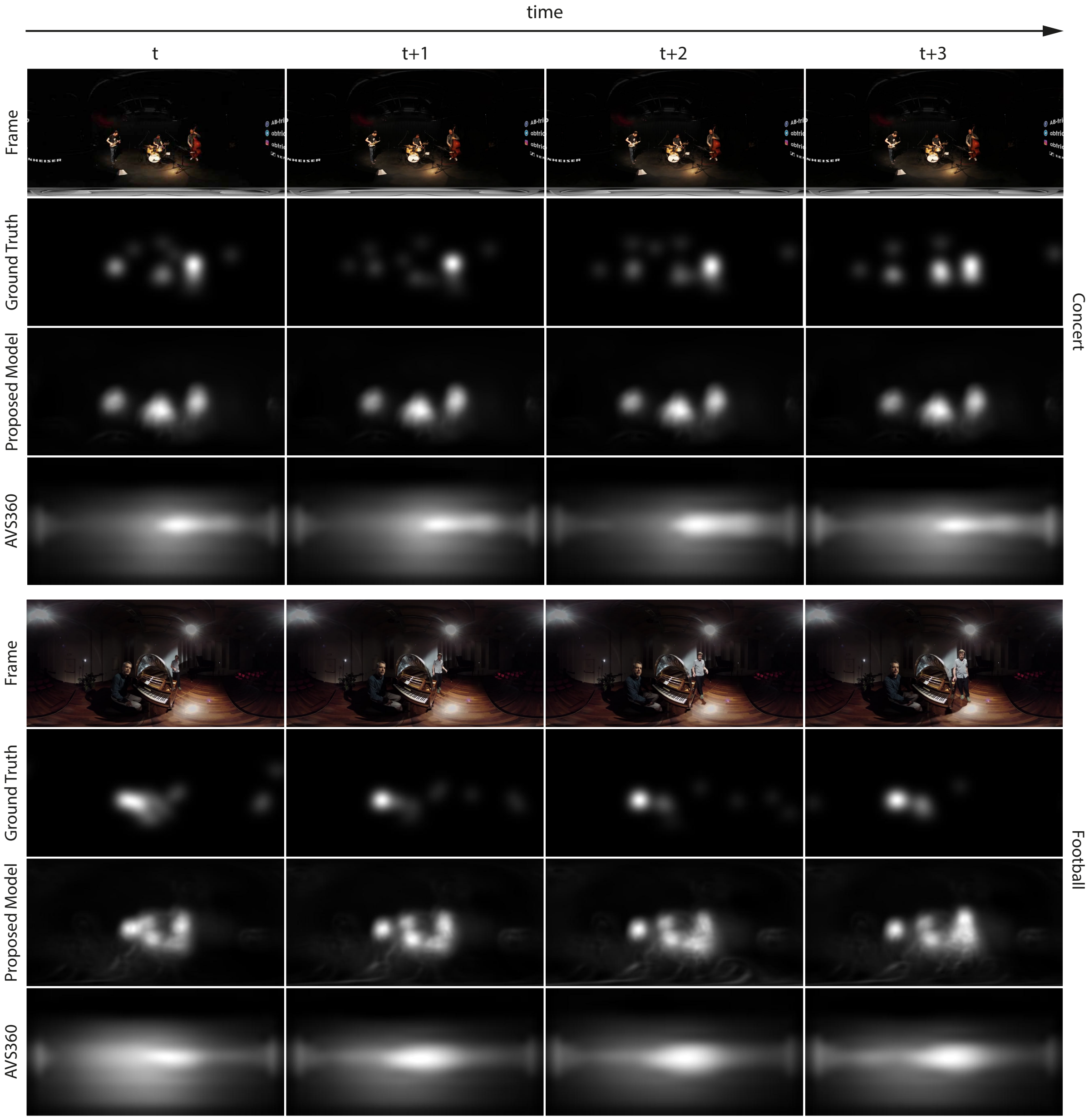

Figure 5.2: Qualitative comparison between the proposed model and AVS360[3] for a series of sequences without longitudinal shifts applied. The horizontal axis represents time. The vertical axis shows from top to bottom: The RGB frame, the ground truth saliency, the proposed model's saliency prediction and AVS360[3] prediction. Black values represent zero probability of being observed and white values high probability. Note that the proposed model outperforms the other technique as the results are visually more similar to the ground truth. The proposed model is able to correctly predict the areas that are more salient producing a fine detailed result with clearly separated salient areas while AVS360[3] produces a more blurry prediction with very large salient areas.

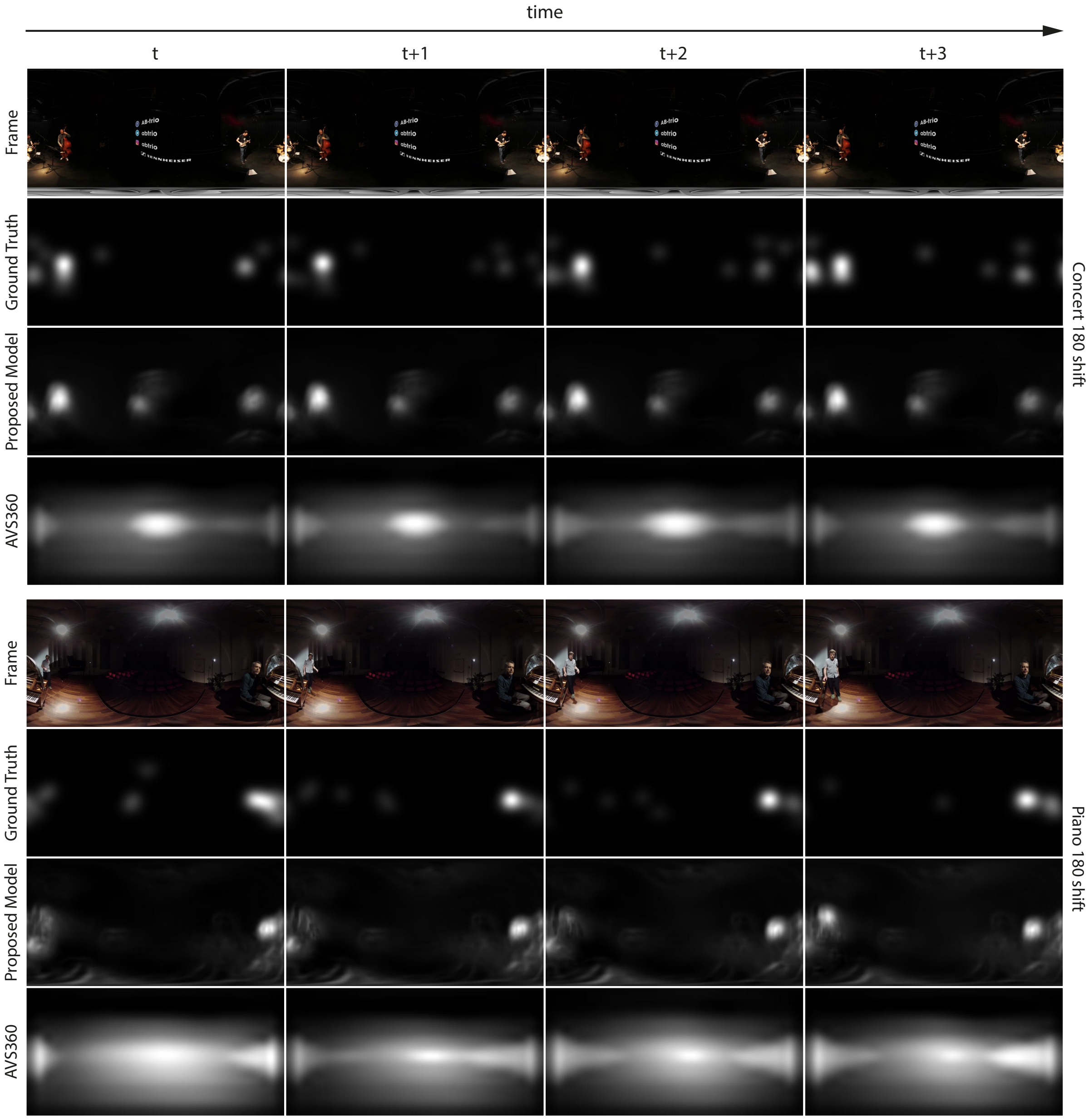

Figure 5.3: Qualitative comparison between the proposed model and AVS360[3] for a series of sequences with a longitudinal shift of 180° applied. The horizontal axis represents time. The vertical axis shows from top to bottom: The RGB frame, the ground truth saliency, the proposed model's saliency prediction and AVS360 prediction. Black values represent zero probability of being observed and white values high probability. The proposed model outperforms the other technique even more clearly than when the sequences had not been longitudinally shifted (Figure 5.2). The proposed model is able to correctly predict the areas that are more salient even if these are far from the center of the image, while AVS360 produces a prediction with very high saliency values towards the center of the image.

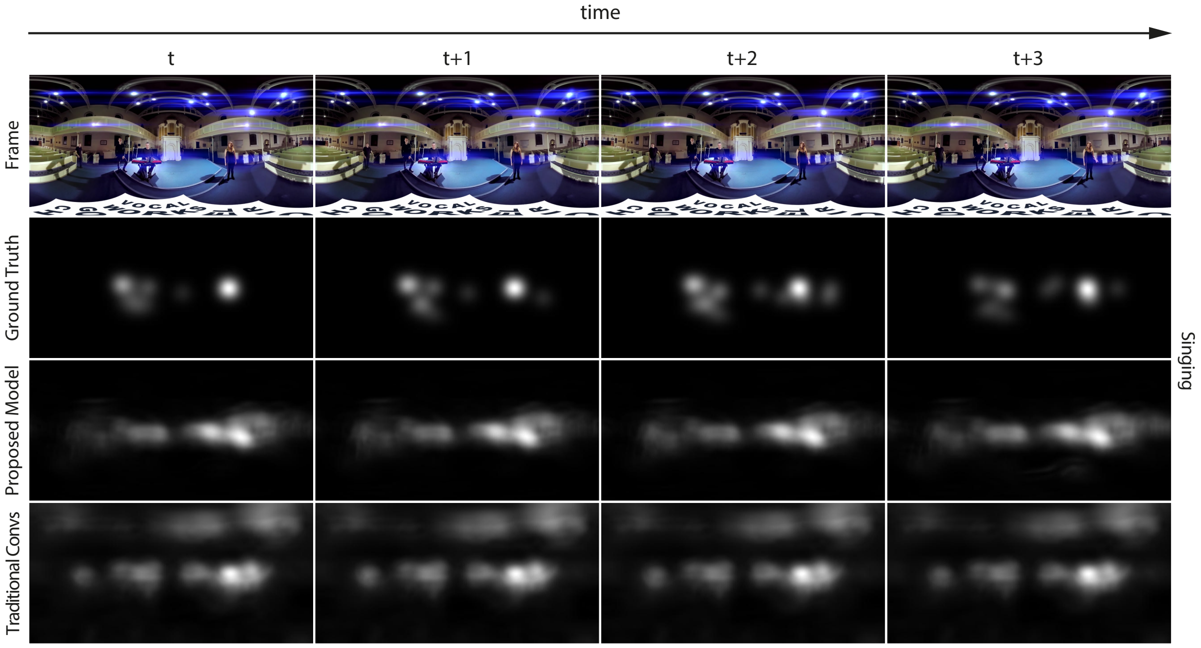

Figure 5.4: Comparison between saliency predictions using Spherical Convolutions and traditional convolutions for a sample sequence. The horizontal axis represents time. The vertical axis shows from top to bottom: The RGB frame, the ground truth saliency, the proposed model's saliency prediction and the prediction of the model using traditional convolutions. Black values represent zero probability of being observed and white values high probability. Note that, while the predictions of both models are very close towards the equator of the image, there are clear differences in the predicted saliency maps towards the poles. The proposed model correctly predicts low values on both poles but the variation with traditional convolutions produces high saliency values on the top of the image that do not match the ground truth saliency map. This is a result of the distortion of the equirectangular projection, which is not accounted for by traditional convolutions and thus affects its results negatively.

6. Conclusions

This work proposes a new deep learning technique for human attention modeling in 360° videos through saliency prediction. The proposed model uses spatio-temporal visual and auditory information, taking into account the distinctive needs related to the chosen equirectangular representation by using 3D Spherical Convolutions based on the ones presented in SphereNet [11]. The proposed model has been evaluated and compared against the state-of-the-art work, producing results that were visually the most accurate and obtained the the best scores across various metrics widely used in the literature.

The results show that Spherical Convolutions are an effective and relatively simple method to overcome the additional difficulties that the spherical projections pose. This method gives the ability to adapt well-studied CNN architectures for 360° content without the need of feeding the model with the same data multiple times through different representations. Additionally, the results of the ablation studies (Section 5.4) continue to support the importance of including temporal information when predicting saliency on 360° videos, as depriving the model of such data greatly decreases the accuracy of the results. More importantly, the work done in this master's thesis shows the positive impact of including both auditory and spatial auditory information on the accuracy of the results obtained.

The proposed solution is able to utilise spatio-temporal auditory information to successfully improve the accuracy of the results obtained. Unlike previous work by Chao et al. [3], the proposed model includes this information without the need for a manually developed attention model. To the best of our knowledge, this is the first work to do so.

6.1 Limitations and future work

In this work, the BCE function was used as the only loss function to train the model although other works such as Jetley et al. [52] or Bruckert et al. [53] show that using the right loss function can greatly increase the accuracy of the results. Further work is needed to study the effects of finding a more suitable loss function on the accuracy of the results.

One of the limitations of the proposed model is that the depth of the temporal information is very limited as it is included in the model through the usage of 3D convolutions and, as such, only short-term temporal information is used. Future work could study the benefits of including longer-term temporal information in the solution by increasing the temporal depth (number of frames fed to the model) or through techniques such as LSTMs or transformers [54].

As mentioned in Section 5.2.1, size of the proposed model is greatly restricted by the memory limits of the GPU used for training. This is due to the current implementation of the 3D Spherical Convolutions, which uses a matrix of sampling positions that take up a significant amount of video memory for each spherical layers in the final proposed model. The current implementation could be further improved by different techniques, such as sharing the sampling positions across different layers when possible (e.g. when the size of input is the same). This would allow implementing larger-sized variations of the proposed model, potentially improving its performance.

Another limitation of the proposed model is caused by the chosen representation of the spatial audio. The audio is fed into the model in two separate parts: a mel-spectrogram and AEMs. While these representations allow the model to easily learn information about characteristics of the audio and the intensities of the audio coming from all directions, these two are learnt separately and they must be merged through a combiner module between the encoder and decoder modules. This supposes a bottleneck in the model and could be limiting its performance. Future work is needed to study alternative representations of the spatial audio that allow the model to learn features about the characteristics of the audio and the directions of the sources in a joined manner.

Lastly, it is worth noticing that the state-of-the-art work, which the proposed model was compared against, uses a different architecture. Although various ablation studies were performed to support the design of the architecture, due to the many differences in the compared architectures, it is hard to draw conclusions about what parts of the model are responsible for the differences observed in the results. Analysis such as using a Resnet architecture in place of U-Net would be greatly beneficial to study how do the differences between the compared methods influence the final results.

Bibliography

[1] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. volume 9351, pages 234–241, 10 2015.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[3] Fang-Yi Chao, Cagri Ozcinar, Lu Zhang, Wassim Hamidouche, Olivier Deforges, and Aljosa Smolic. Towards audio-visual saliency prediction for omnidirectional video with spatial audio. In *2020 IEEE International Conference on Visual Communications and Image Processing (VCIP)*, pages 355–358, 2020.

[4] 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.

[5] Laurent Itti and Christof Koch. A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision Research*, 40(10):1489–1506, 2000.

[6] Neil Bruce and John Tsotsos. Saliency based on information maximization. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems*, volume 18. MIT Press, 2005.

[7] Jonathan Harel, Christof Koch, and Pietro Perona. Graph-based visual saliency. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2006.

[8] Junting Pan, Cristian Canton, Kevin McGuinness, Noel O'Connor, Jordi Torres, Elisa Sayrol, and Xavier Giró-i Nieto. Salgan: Visual saliency prediction with generative adversarial networks. 01 2017.

[9] Alexander Kroner, Mario Senden, Kurt Driessens, and Rainer Goebel. Contextual encoder-decoder network for visual saliency prediction. *Neural Networks*, 129, 05 2020.

[10] Fang-Yi Chao, Lu Zhang, Wassim Hamidouche, and Olivier Déforges. Salgan360: Visual saliency prediction on 360 degree images with generative adversarial networks. 07 2018.

[11] Benjamin Coors, Alexandru Paul Condurache, and Andreas Geiger. *SphereNet: Learning Spherical Representations for Detection and Classification in Omnidirectional Images: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part IX*, pages 525–541. 09 2018.

[12] Hsien-Tzu Cheng, Chun-Hung Chao, Jin-Dong Dong, Hao-Kai Wen, Tyng-Luh Liu, and Min Sun. Cube padding for weakly-supervised saliency prediction in 360° videos. pages 1420–1429, 06 2018.

[13] 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.

[14] Ziheng Zhang, Yanyu Xu, Jingyi Yu, and Shenghua Gao. Saliency detection in 360° videos. In *ECCV 2018*, 2018.

[15] Yanyu Xu, Ziheng Zhang, and Shenghua Gao. Spherical dnns and their applications in 360° images and videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2021.

[16] Fang-Yi Chao, Cagri Ozcinar, Chen Wang, Emin Zerman, Lu Zhang, Wassim Hamidouche, Olivier Déforges, and Aljosa Smolic. Audio-visual perception of omnidirectional video for virtual reality applications. 06 2020.

[17] Antoine Coutrot, Nathalie Guyader, Gelu Ionescu, and Alice Caplier. Influence of soundtrack on eye movements during video exploration. *Journal of Eye Movement Research*, 5:1–10, 08 2012.

[18] Anna Vilaro, Andrew Duchowski, Pilar Orero, Tom Grindinger, Stephen Tetreault, and Elena Giovanni. How sound is the pear tree story? testing the effect of varying audio stimuli on visual attention distribution. *Perspectives-studies in Translatology*, 20:55–65, 03 2012.

[19] Jeremy Wolfe. Guided search 2.0 a revised model of visual search. *Psychonomic Bulletin Review*, 1:202–238, 06 1994.

[20] Erdan Gu, Jingbin Wang, and N.I. Badler. Generating sequence of eye fixations using decision-theoretic attention model. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*, pages 92–92, 2005.

[21] 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.

[22] Eleonora Vig, Michael Dorr, and David Cox. Large-scale optimization of hierarchical features for saliency prediction in natural images. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2798–2805, 2014.

[23] Srinivas Kruthiventi, Kumar Ayush, and R. Babu. Deepfix: A fully convolutional neural network for predicting human eye fixations. *IEEE Transactions on Image Processing*, PP, 10 2015.

[24] Junting Pan, Kevin McGuinness, Elisa Sayrol, Noel O'Connor, and Xavier Giró-i Nieto. Shallow and deep convolutional networks for saliency prediction. 03 2016.

[25] Nian Liu and Junwei Han. A deep spatial contextual long-term recurrent convolutional network for saliency detection. *IEEE Transactions on Image Processing*, PP, 10 2016.

[26] Dashan Gao, Vijay Mahadevan, and Nuno Vasconcelos. The discriminant center-surround hypothesis for bottom-up saliency. volume 20, 01 2007.

[27] Hae Seo and Peyman Milanfar. Using local regression kernels for statistical object detection. pages 2380–2383, 01 2008.

[28] Çağdaş Bak, Aysun Koçak, Erkut Erdem, and Aykut Erdem. Spatio-temporal saliency networks for dynamic saliency prediction. *IEEE Transactions on Multimedia*, PP:1–1, 11 2017.

[29] George Leifman, Dmitry Rudoy, Tristan Swedish, Eduardo Bayro-Corrochano, and Ramesh Raskar. Learning gaze transitions from depth to improve video saliency estimation. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1707–1716, 2017.

[30] Siavash Gorji and James Clark. Going from image to video saliency: Augmenting image salience with dynamic attentional push. pages 7501–7511, 06 2018.

[31] Guanghan Song. Effect of sound in videos on gaze : contribution to audio-visual saliency modelling. 06 2013.

[32] Naty Sidaty, Chaker Larabi, and Hakim Saadane. An audiovisual saliency model for conferencing and conversation videos. volume 2016, 02 2016.

[33] Hamed Rezazadegan Tavakoli, Ali Borji, Esa Rahtu, and Juho Kannala. Dave: A deep audio-visual embedding for dynamic saliency prediction, 05 2019.

[34] Yuming Fang, Xiaoqiang Zhang, and Nevrez imamoğlu. A novel superpixel-based saliency detection model for 360-degree images. *Signal Processing: Image Communication*, 69, 08 2018.

[35] Rafael Monroy, Sebastian Lutz, Tejo Chalasani, and Aljosa Smolic. Salnet360: Saliency maps for omni-directional images with cnn. *Signal Processing: Image Communication*, 69, 09 2017.

[36] Joonatan Mänttäri, Sofia Broomé, John Folkesson, and Hedvig Kjellström. Interpreting video features: a comparison of 3d convolutional networks and convolutional LSTM networks. *CoRR*, abs/2002.00367, 2020.

[37] Fei Yan, Cheng Chen, Peng Xiao, Siyu Qi, Zhiliang Wang, and Ruoxiu Xiao. Review of visual saliency prediction: Development process from neurobiological basis to deep models. *Applied Sciences*, 12(1), 2022.

[38] L. Itti. Automatic foveation for video compression using a neurobiological model of visual attention. *IEEE Transactions on Image Processing*, 13(10):1304–1318, 2004.

[39] Ke Wang, Sai Ma, Fan Ren, and Jianbo Lu. Sbas: Salient bundle adjustment for visual slam. *IEEE Transactions on Instrumentation and Measurement*, 70:1–9, 2021.

[40] Ekrem Aksoy, Ahmet Yazıcı, and Mahmut Kasap. See, attend and brake: An attention-based saliency map prediction model for end-to-end driving, 02 2020.

[41] Shuo Wang, Ming Jiang, Xavier Morin Duchesne, Elizabeth A Laugeson, Daniel P Kennedy, Ralph Adolphs, and Qi Zhao. Atypical visual saliency in autism spectrum disorder quantified through model-based eye tracking. *Neuron*, pages –, 10 2015.

[42] 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.

[43] michael a. gerzon. the design of precisely coincident microphone arrays for stereo and surround sound. *journal of the audio engineering society*, march 1975.

[44] Apple. `https://support.apple.com/ar-ae/guide/logicpro-iru/dev022fbc493/mac`.

[45] Pedro Morgado, Nuno Nvasconcelos, Timothy Langlois, and Oliver Wang. Self-supervised generation of spatial audio for 360°video. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[46] Anh H. Reynolds. `https://anhreynolds.com/blogs/cnn.html`.

[47] Manuel Danner. Cell segmentation with the u-net convolutional network. 2020.

[48] Pavel Tokmakov, Alahari Karteek, and Cordelia Schmid. Learning motion patterns in videos. In *CVPR*, 2017.

[49] Yi Zhang, Fang-Yi Chao, Ge-Peng Ji, Deng-Ping Fan, Lu Zhang, and Ling Shao. Asod60k: Audio-induced salient object detection in panoramic videos, 07 2021.

[50] Robert J. Peters, Asha Iyer, Laurent Itti, and Christof Koch. Components of bottom-up gaze allocation in natural images. *Vision Research*, 45(18):2397–2416, 2005.

[51] Jesús Gutiérrez, Erwan David, Yashas Rai, and Patrick Le Callet. Toolbox and dataset for the development of saliency and scanpath models for omnidirectional/360° still images. *Signal Processing: Image Communication*, 69, 05 2018.

[52] Saumya Jetley, Naila Murray, and Eleonora Vig. End-to-end saliency mapping via probability distribution prediction. *CoRR*, abs/1804.01793, 2018.

[53] Alexandre Bruckert, Hamed R. Tavakoli, Zhi Liu, Marc Christie, and Olivier Le Meur. Deep saliency models : The quest for the loss function, 2019.

[54] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

Appendix A. Evaluation metrics

This appendix presents a detailed definition of the different evaluation metrics used for the evaluation of the proposed model (Section 5.1):

Correlation Coefficient (CC): it is a statistical function that measures the dependence or linear correlation between two variables, in the case of saliency prediction, the predicted and ground truth saliency maps. The metric takes a value equal to 0 when the two saliency maps are independent (not correlated), a value equal to 1 when the maps are perfectly correlated, and a value of -1 when the maps are perfectly correlated but opposite. First, the saliency maps are standardized to have zero mean and unit standard deviation; then, the metric is computed as follows:

$$CC(P,G) = \frac{cov\left(\tilde{P},\tilde{G}\right)}{\sigma\left(\tilde{P}\right)\sigma\left(\tilde{G}\right)} \tag{A.1}$$

where $cov\left(\tilde{P},\tilde{G}\right)$ is the covariance of the two saliency maps; $\sigma\left(\tilde{P}\right)$ and $\sigma\left(\tilde{G}\right)$ are the standard deviation of the predicted and ground truth saliency maps respectively.

Normalized Scanpath Saliency (NSS) [50]: it is a metric that takes a saliency map and a set of fixations as input and measured the values of saliency at the fixation locations. First, the saliency map is standardized to have zero mean and unit standard deviation; then, the value of the resulting map is measured at each fixation point and the mean is computed:

$$NSS(\tilde{P},F) = \frac{1}{M}\sum_{f\in F}\tilde{P}\left(x_f\right) \tag{A.2}$$

where M is the number of total fixations F; $\tilde{P}\left(x_f\right)$ is the value of normalized predicted saliency at the position of fixation f; μ is the mean of the predicted saliency map; and σ is the standard deviation of the predicted saliency map.

Kullback-Leibler Divergence (KLD): it is a statistical function that measures the overall dissimilarity between two probability density functions, in this case, the ground truth and predicted saliency maps. As such, the two saliency maps are interpreted as probability distributions and normalized so the sum of all the pixels in each image is equal to one. The metric has its lowest value at 0, where the two saliency maps are equal; and the highest value at infinity.

$$KLD(\hat{P},\hat{G}) = \sum_{x \in X} \hat{G}(x) \log \left(\epsilon + \frac{\hat{G}\left(x\right)}{\epsilon + \hat{P}\left(x\right)} \right) \tag{A.3}$$

where $\hat{P}$ and $\hat{G}$ are the normalized predicted and ground truth saliency maps respectively; X is the image space; $\hat{G}\left(x\right)$ and $\hat{P}\left(x\right)$ are the normalized saliency values of the ground truth and prediction at position x respectively; and ϵ is a small constant to avoid divisions by zero.

Similarity Metric (SIM): it is a function that measures the amount of overlap (and thus overall similarity) between the predicted and ground truth saliency maps, interpreted as probability density functions. The metric has its lowest value at 0, when the two saliency maps do not overlap; and the highest value at one, when both saliency maps are equal.

$$SIM(\hat{P},\hat{G}) = \sum_{x \in X} min \left(\hat{P}(x), \hat{G}(x) \right) \tag{A.4}$$

Bhattacharyya coefficient (BC): it is a function that, as the SIM metric, measures the overall similarity between the predicted and ground truth saliency maps, interpreted as probability density functions. The metric has its lowest value at 0, when the two saliency maps do not overlap; and the highest value at one, when both saliency maps are equal.

$$BC(\hat{P},\hat{G}) = \sum_{x \in X} \sqrt{\hat{P}(x)\hat{G}(x)} \tag{A.5}$$

Appendix B. Quantitative evaluation

This appendix presents detailed results obtained in the quantitative evaluation of the proposed model against AVS360[3] (Section 5.3 and the ablation studies performed (Section 5.4). The metrics used for the evaluation are detailed in Appendix A. The average results obtained for all the frames of each video are shown for the different evaluated models.

The videos in the used dataset, ASOD60k[49], present many different real-world scenarios (e.g., sports, concerts, interviews, etc.) and different motion patterns (e.g., static/moving camera). This might cause the performance of the different models to vary notably from one video to another. Moreover, the dataset was augmented by applying a set of longitudinal shifts for the comparison study against the state-of-the-art, which might cause the performance of the model to change across the different variations of the same sequence. Thus, the performance of a model cannot be evaluated by only accounting for the average results across all videos but the individual results should be studied. For this reason, this appendix provides the results of the CC, NSS, SIM, BC, and KLD metrics for each video, showing the mean and the standard deviation across all frames.

Comparison with state of the art. Section 5.3 presents a comparison of the proposed model against a state-of-the-art work, AVS360[3]. The results for each video of the augmented dataset with the proposed model and AVS360 are shown in Tables B.1 and B.2.

Ablation studies. Section 5.4 presents a series of ablation studies to support the decisions made during the design of the proposed model. The results obtained for all videos with each of the models implemented in the different ablation studies are shown in Tables B.3, B.4, B.5, B.6 and B.7. Note that the non augmented ASOD60k dataset was used for the evaluation of the different models.

	CC		NSS		SIM		BC		KLD	
video	mean	std	mean	std	mean	std	mean	std	mean	std
1	0.528	0.09	2.256	0.591	0.411	0.077	0.651	0.083	10.896	3.802
1(90° shift)	0.542	0.095	2.395	0.652	0.43	0.078	0.668	0.082	10.261	3.788
1(180° shift)	0.481	0.104	1.981	0.593	0.375	0.081	0.616	0.088	12.124	3.816
1(270° shift)	0.493	0.126	2.04	0.529	0.387	0.078	0.624	0.081	11.839	3.262
2	0.203	0.055	0.887	0.238	0.301	0.034	0.535	0.043	13.108	2.039
2(90° shift)	0.188	0.078	0.988	0.298	0.298	0.033	0.517	0.044	13.116	2.013
2(180° shift)	0.209	0.071	0.995	0.315	0.3	0.044	0.528	0.054	13.756	2.212
2(270° shift)	0.212	0.06	0.86	0.253	0.299	0.041	0.533	0.048	13.823	2.034
3	0.415	0.176	1.48	0.575	0.44	0.08	0.704	0.064	5.055	1.084
3(90° shift)	0.414	0.157	1.429	0.483	0.444	0.064	0.706	0.054	5.972	1.473
3(180° shift)	0.415	0.133	1.454	0.456	0.44	0.053	0.705	0.049	6.618	1.733
3(270° shift)	0.448	0.149	1.603	0.471	0.456	0.063	0.724	0.05	5.411	1.313
4	0.493	0.114	1.717	0.515	0.373	0.065	0.607	0.061	10.652	3.23
4(90° shift)	0.414	0.156	1.369	0.61	0.326	0.065	0.554	0.068	12.695	2.676
4(180° shift)	0.427	0.128	1.414	0.556	0.34	0.064	0.576	0.071	12.566	3.44
4(270° shift)	0.378	0.142	1.323	0.529	0.358	0.079	0.587	0.074	11.045	4.019
5	0.391	0.171	1.994	1.063	0.346	0.08	0.591	0.084	11.837	3.064
5(90° shift)	0.37	0.19	1.9	0.911	0.342	0.1	0.589	0.104	11.682	3.823
5(180° shift)	0.386	0.257	2.011	1.262	0.362	0.133	0.602	0.139	10.404	4.3
5(270° shift)	0.368	0.255	1.967	1.389	0.362	0.126	0.604	0.128	10.391	4.439
6	0.438	0.12	1.798	0.528	0.396	0.075	0.65	0.08	9.171	3.186
6(90° shift)	0.436	0.136	1.831	0.543	0.405	0.08	0.655	0.091	9.284	3.459
6(180° shift)	0.317	0.148	1.353	0.5	0.354	0.09	0.59	0.107	10.925	3.913
6(270° shift)	0.5	0.118	2.014	0.441	0.396	0.07	0.64	0.074	10.572	2.609
7	0.275	0.175	0.697	0.562	0.34	0.085	0.584	0.104	9.93	3.862
7(90° shift)	0.239	0.145	0.617	0.474	0.325	0.074	0.562	0.095	11.557	3.579
7(180° shift)	0.254	0.161	0.693	0.552	0.312	0.074	0.546	0.097	13.039	3.635
7(270° shift)	0.292	0.205	0.751	0.626	0.338	0.087	0.574	0.102	10.765	3.628
8	0.448	0.078	2.441	0.681	0.31	0.05	0.548	0.055	14.837	2.312
8(90° shift)	0.356	0.074	1.877	0.568	0.297	0.052	0.53	0.061	14.935	2.819
8(180° shift)	0.084	0.041	0.541	0.37	0.176	0.057	0.373	0.077	18.901	3.147
8(270° shift)	0.29	0.088	1.523	0.475	0.242	0.054	0.467	0.071	17.157	2.824
9	0.541	0.064	1.915	0.258	0.47	0.06	0.723	0.061	7.285	2.713
9(90° shift)	0.474	0.071	1.721	0.276	0.432	0.066	0.686	0.065	8.407	2.955
9(180° shift)	0.227	0.084	0.985	0.327	0.341	0.069	0.58	0.082	11.23	3.586
9(270° shift)	0.388	0.067	1.393	0.296	0.387	0.06	0.641	0.07	9.998	3.384
10	0.418	0.251	1.565	0.889	0.36	0.124	0.599	0.135	10.765	3.856
10(90° shift)	0.284	0.273	1.013	0.971	0.279	0.129	0.494	0.157	14.382	4.544
10(180° shift)	0.046	0.077	0.213	0.342	0.195	0.072	0.387	0.101	16.842	4.278
10(270° shift)	0.265	0.253	0.894	0.891	0.281	0.132	0.494	0.158	14.88	5.075
11	0.679	0.07	3.02	0.442	0.526	0.054	0.775	0.041	4.837	1.101
11(90° shift)	0.671	0.099	2.818	0.569	0.517	0.06	0.764	0.048	5.665	1.318
11(180° shift)	0.64	0.17	2.662	0.7	0.496	0.075	0.729	0.065	7.316	1.809
11(270° shift)	0.593	0.055	2.298	0.331	0.486	0.047	0.741	0.04	5.497	1.306
12	0.58	0.111	2.945	0.483	0.395	0.056	0.638	0.061	11.525	2.365
12(90° shift)	0.461	0.142	2.24	0.553	0.336	0.062	0.58	0.08	13.31	3.053
12(180° shift)	0.163	0.081	0.965	0.459	0.222	0.039	0.444	0.053	16.842	2.688
12(270° shift)	0.583	0.106	2.806	0.535	0.376	0.052	0.616	0.056	12.496	1.782
13	0.489	0.141	2.496	1.122	0.335	0.054	0.557	0.06	12.956	3.438
13(90° shift)	0.612	0.17	3.165	1.433	0.408	0.058	0.646	0.049	10.927	2.505
13(180° shift)	0.218	0.209	1.167	0.904	0.25	0.114	0.461	0.143	16.69	4.48
13(270° shift)	0.37	0.077	1.673	0.522	0.308	0.064	0.544	0.08	14.185	4.149
14	0.496	0.133	2.6	0.446	0.382	0.1	0.632	0.099	10.895	3.859
14(90° shift)	0.508	0.161	2.536	0.848	0.391	0.099	0.637	0.101	10.886	3.829
14(180° shift)	0.535	0.099	2.839	0.597	0.341	0.077	0.577	0.081	13.111	2.952
14(270° shift)	0.537	0.109	2.818	0.61	0.369	0.086	0.618	0.083	11.333	2.935

Table B.1: Mean and standard deviation of the CC, NSS, SIM, BC, and KLD metrics for each video in the augmented ASOD60k dataset for the evaluation of the proposed model.

	CC		NSS		SIM		BC		KLD	
video	mean	std	mean	std	mean	std	mean	std	mean	std
1	0.563	0.096	2.407	0.467	0.43	0.066	0.676	0.067	9.695	2.717
1(90° shift)	0.495	0.099	2.142	0.473	0.389	0.077	0.631	0.083	11.588	3.457
1(180° shift)	0.34	0.097	1.424	0.407	0.315	0.083	0.554	0.094	14.152	4.058
1(270° shift)	0.481	0.138	2.032	0.419	0.381	0.082	0.62	0.087	11.832	3.224
2	0.462	0.076	1.677	0.227	0.42	0.04	0.675	0.043	8.953	1.696
2(90° shift)	0.223	0.071	0.885	0.22	0.346	0.042	0.584	0.053	10.568	2.184
2(180° shift)	0.31	0.066	1.145	0.273	0.357	0.055	0.611	0.064	10.769	2.948
2(270° shift)	0.458	0.062	1.599	0.214	0.406	0.051	0.662	0.055	9.623	2.39
3	0.423	0.183	1.4	0.576	0.453	0.069	0.693	0.066	6.913	1.625
3(90° shift)	0.449	0.12	1.447	0.364	0.453	0.048	0.706	0.049	7.609	1.754
3(180° shift)	0.466	0.06	1.395	0.217	0.444	0.043	0.701	0.044	8.327	1.946
3(270° shift)	0.529	0.114	1.707	0.333	0.492	0.049	0.741	0.045	6.629	1.78
4	0.55	0.085	1.81	0.233	0.463	0.046	0.707	0.047	8.49	1.81
4(90° shift)	0.541	0.082	1.725	0.256	0.439	0.042	0.684	0.043	9.631	1.684
4(180° shift)	0.449	0.072	1.481	0.254	0.382	0.042	0.628	0.048	11.661	2.194
4(270° shift)	0.469	0.08	1.589	0.264	0.411	0.045	0.655	0.048	10.447	2.094
5	0.393	0.111	1.857	0.608	0.348	0.063	0.596	0.071	11.977	2.968
5(90° shift)	0.378	0.139	1.746	0.532	0.338	0.077	0.581	0.089	12.524	3.234
5(180° shift)	0.371	0.123	1.837	0.673	0.332	0.063	0.574	0.069	12.531	2.568
5(270° shift)	0.368	0.125	1.803	0.749	0.339	0.054	0.589	0.067	11.875	2.764
6	0.408	0.105	1.461	0.324	0.385	0.065	0.639	0.071	9.937	2.504
6(90° shift)	0.397	0.123	1.498	0.35	0.386	0.073	0.634	0.082	9.953	2.803
6(180° shift)	0.333	0.136	1.332	0.419	0.339	0.072	0.573	0.084	12.423	2.749
6(270° shift)	0.366	0.094	1.345	0.285	0.344	0.057	0.586	0.066	12.569	2.262
7	0.301	0.161	0.624	0.456	0.363	0.084	0.597	0.105	10.56	3.57
7(90° shift)	0.223	0.149	0.455	0.4	0.334	0.084	0.561	0.11	11.914	3.798
7(180° shift)	0.111	0.117	0.201	0.324	0.282	0.077	0.495	0.108	13.752	3.914
7(270° shift)	0.237	0.144	0.475	0.386	0.333	0.084	0.56	0.106	12.185	3.687
8	0.58	0.061	3.118	0.395	0.354	0.065	0.597	0.067	13.266	2.481
8(90° shift)	0.559	0.07	2.98	0.425	0.329	0.065	0.566	0.069	14.246	2.491
8(180° shift)	0.207	0.063	1.177	0.337	0.219	0.071	0.425	0.084	17.932	3.095
8(270° shift)	0.472	0.064	2.515	0.456	0.3	0.059	0.536	0.065	15.328	2.502
9	0.613	0.058	2.137	0.31	0.468	0.056	0.716	0.056	8.162	2.382
9(90° shift)	0.5	0.085	1.708	0.232	0.418	0.067	0.668	0.072	9.723	2.938
9(180° shift)	0.239	0.073	0.92	0.273	0.34	0.066	0.575	0.079	12.082	3.421
9(270° shift)	0.502	0.08	1.741	0.385	0.414	0.055	0.663	0.06	10.255	2.704
10	0.495	0.084	1.841	0.404	0.395	0.063	0.635	0.073	11.024	2.715
10(90° shift)	0.386	0.067	1.467	0.453	0.353	0.06	0.592	0.075	12.291	3.146
10(180° shift)	0.202	0.077	0.796	0.319	0.281	0.068	0.509	0.091	14.372	3.8
10(270° shift)	0.393	0.088	1.462	0.345	0.353	0.066	0.593	0.082	12.296	3.289
11	0.551	0.062	1.888	0.29	0.415	0.035	0.666	0.037	10.095	1.377
11(90° shift)	0.417	0.063	1.411	0.232	0.363	0.041	0.61	0.046	12.141	1.964
11(180° shift)	0.243	0.066	0.809	0.208	0.304	0.047	0.545	0.059	13.841	2.591
11(270° shift)	0.438	0.073	1.506	0.25	0.372	0.045	0.62	0.051	11.733	2.089
12	0.358	0.054	1.751	0.236	0.297	0.038	0.531	0.047	14.736	2.154
12(90° shift)	0.248	0.075	1.247	0.317	0.258	0.051	0.487	0.066	15.905	2.813
12(180° shift)	0.137	0.036	0.756	0.212	0.222	0.044	0.431	0.05	17.272	2.357
12(270° shift)	0.319	0.063	1.552	0.29	0.27	0.032	0.498	0.037	16.097	1.445
13	0.17	0.127	0.754	0.59	0.229	0.089	0.421	0.124	17.048	3.772
13(90° shift)	0.149	0.138	0.605	0.541	0.229	0.097	0.425	0.134	16.768	4.33
13(180° shift)	0.107	0.165	0.425	0.678	0.215	0.109	0.397	0.162	16.977	4.936
13(270° shift)	0.108	0.12	0.491	0.484	0.214	0.09	0.404	0.127	17.294	4.237
14	0.49	0.136	2.286	0.568	0.328	0.067	0.569	0.077	13.66	2.612
14(90° shift)	0.393	0.129	1.852	0.508	0.284	0.063	0.518	0.078	15.362	2.62
14(180° shift)	0.252	0.06	1.152	0.316	0.223	0.045	0.445	0.059	17.649	2.192
14(270° shift)	0.443	0.074	2.025	0.412	0.29	0.05	0.528	0.061	15.295	2.147

Table B.2: Mean and standard deviation of the CC, NSS, SIM, BC, and KLD metrics for each video in the augmented ASOD60k dataset for the evaluation of AVS360[3].

	CC		NSS		SIM		BC		KLD	
video	mean	std	mean	std	mean	std	mean	std	mean	std
1	0.528	0.09	2.256	0.591	0.411	0.077	0.651	0.083	10.896	3.802
2	0.203	0.055	0.887	0.238	0.301	0.034	0.535	0.043	13.108	2.039
3	0.415	0.176	1.48	0.575	0.44	0.08	0.704	0.064	5.055	1.084
4	0.493	0.114	1.717	0.515	0.373	0.065	0.607	0.061	10.652	3.23
5	0.391	0.171	1.994	1.063	0.346	0.08	0.591	0.084	11.837	3.064
6	0.438	0.12	1.798	0.528	0.396	0.075	0.65	0.08	9.171	3.186
7	0.275	0.175	0.697	0.562	0.34	0.085	0.584	0.104	9.93	3.862
8	0.448	0.078	2.441	0.681	0.31	0.05	0.548	0.055	14.837	2.312
9	0.541	0.064	1.915	0.258	0.47	0.06	0.723	0.061	7.285	2.713
10	0.418	0.251	1.565	0.889	0.36	0.124	0.599	0.135	10.765	3.856
11	0.679	0.07	3.02	0.442	0.526	0.054	0.775	0.041	4.837	1.101
12	0.58	0.111	2.945	0.483	0.395	0.056	0.638	0.061	11.525	2.365
13	0.489	0.141	2.496	1.122	0.335	0.054	0.557	0.06	12.956	3.438
14	0.496	0.133	2.6	0.446	0.382	0.1	0.632	0.099	10.895	3.859

Table B.3: Mean and standard deviation of the CC, NSS, SIM, BC, and KLD metrics for each video in the ASOD60k dataset for the evaluation of the proposed model.

	CC		NSS		SIM		BC		KLD	
video	mean	std	mean	std	mean	std	mean	std	mean	std
1	0.489	0.111	2.207	0.43	0.386	0.076	0.62	0.077	11.954	3.183
2	0.308	0.061	1.216	0.252	0.362	0.026	0.607	0.034	10.985	1.56
3	0.158	0.15	0.538	0.541	0.284	0.069	0.526	0.076	9.981	1.707
4	0.304	0.127	1.056	0.474	0.326	0.077	0.576	0.088	11.77	4.22
5	0.462	0.121	2.224	0.561	0.368	0.082	0.61	0.082	11.856	3.192
6	0.42	0.089	1.614	0.322	0.37	0.059	0.615	0.067	11.512	2.32
7	0.308	0.183	0.778	0.624	0.351	0.072	0.596	0.09	11.173	3.212
8	0.134	0.061	0.801	0.376	0.174	0.063	0.374	0.082	19.043	3.145
9	0.485	0.063	1.706	0.308	0.391	0.052	0.635	0.057	10.952	2.211
10	0.493	0.081	1.786	0.279	0.398	0.07	0.644	0.077	10.472	2.861
11	0.614	0.092	2.722	0.499	0.399	0.04	0.636	0.039	11.643	1.186
12	0.438	0.167	2.374	0.618	0.319	0.071	0.552	0.101	13.732	3.299
13	0.31	0.163	1.322	0.699	0.301	0.1	0.542	0.124	12.205	5.85
14	0.438	0.103	2.357	0.409	0.278	0.063	0.512	0.072	15.63	2.46

Table B.4: Mean and standard deviation of the CC, NSS, SIM, BC, and KLD metrics for each video in the ASOD60k dataset for the evaluation of the model implemented in Section 5.4.1.

	CC		NSS		SIM		BC		KLD	
video	mean	std	mean	std	mean	std	mean	std	mean	std
1	0.572	0.154	2.529	0.537	0.449	0.086	0.69	0.074	9.07	2.769
2	0.267	0.067	1.139	0.279	0.321	0.027	0.552	0.037	13.402	1.566
3	0.271	0.118	0.97	0.382	0.339	0.042	0.566	0.049	10.668	1.514
4	0.313	0.231	1.067	0.732	0.349	0.12	0.581	0.124	11.804	4.682
5	0.34	0.195	1.718	1.123	0.31	0.085	0.554	0.093	12.545	3.114
6	0.314	0.116	1.322	0.475	0.335	0.068	0.579	0.08	10.682	3.815
7	0.21	0.159	0.545	0.514	0.303	0.069	0.54	0.089	12.903	3.255
8	0.373	0.106	1.853	0.906	0.259	0.051	0.491	0.06	16.013	2.154
9	0.42	0.076	1.478	0.251	0.392	0.06	0.642	0.066	10.166	2.57
10	0.781	0.069	3.141	0.815	0.531	0.071	0.762	0.058	7.11	2.336
11	0.502	0.085	1.966	0.409	0.367	0.036	0.616	0.039	11.65	1.315
12	0.446	0.192	1.925	0.743	0.316	0.077	0.544	0.106	12.706	2.813
13	0.249	0.249	1.227	0.972	0.268	0.129	0.477	0.162	15.361	5.267
14	0.339	0.187	1.712	0.868	0.266	0.076	0.495	0.096	15.486	3.169

Table B.5: Mean and standard deviation of the CC, NSS, SIM, BC, and KLD metrics for each video in the ASOD60k dataset for the evaluation of the model implemented in Section 5.4.2.

	CC		NSS		SIM		BC		KLD	
video	mean	std	mean	std	mean	std	mean	std	mean	std
1	0.406	0.109	1.651	0.366	0.339	0.074	0.575	0.083	13.907	3.508
2	0.328	0.072	1.474	0.254	0.348	0.027	0.583	0.032	12.182	1.271
3	0.21	0.118	0.714	0.4	0.321	0.065	0.567	0.083	7.078	1.66
4	0.3	0.147	1.117	0.608	0.323	0.079	0.567	0.091	11.936	4.633
5	0.449	0.163	2.274	1.0	0.359	0.084	0.599	0.087	12.032	3.091
6	0.504	0.121	2.131	0.596	0.426	0.069	0.679	0.07	8.203	2.93
7	0.27	0.15	0.671	0.53	0.343	0.078	0.591	0.101	10.509	3.731
8	0.36	0.079	1.857	0.648	0.246	0.042	0.471	0.05	16.917	1.762
9	0.434	0.047	1.46	0.271	0.389	0.046	0.643	0.05	9.853	2.051
10	0.62	0.093	2.359	0.451	0.447	0.076	0.685	0.077	9.674	2.793
11	0.261	0.044	0.916	0.251	0.27	0.034	0.5	0.042	15.477	1.422
12	0.565	0.227	2.914	0.91	0.389	0.086	0.614	0.113	11.816	2.615
13	0.3	0.208	1.385	0.782	0.287	0.115	0.51	0.138	14.897	4.984
14	0.302	0.117	1.748	0.449	0.223	0.06	0.443	0.079	17.19	2.485

Table B.6: Mean and standard deviation of the CC, NSS, SIM, BC, and KLD metrics for each video in the ASOD60k dataset for the evaluation of the model implemented in Section 5.4.3.

	CC		NSS		SIM		BC		KLD	
video	mean	std	mean	std	mean	std	mean	std	mean	std
1	0.576	0.099	2.717	0.831	0.358	0.062	0.591	0.068	13.891	2.635
2	0.348	0.086	1.439	0.252	0.39	0.036	0.631	0.044	9.946	1.662
3	0.37	0.168	1.387	0.537	0.41	0.054	0.664	0.054	7.844	1.428
4	0.454	0.073	1.566	0.284	0.419	0.05	0.659	0.047	10.256	2.084
5	0.565	0.168	3.042	1.011	0.413	0.096	0.651	0.089	10.438	3.108
6	0.556	0.125	2.214	0.665	0.468	0.067	0.726	0.057	6.238	2.239
7	0.351	0.149	0.872	0.533	0.392	0.081	0.64	0.093	8.567	3.385
8	0.159	0.076	1.045	0.492	0.207	0.073	0.409	0.088	18.191	3.024
9	0.28	0.071	1.035	0.272	0.328	0.058	0.569	0.069	12.353	2.597
10	0.507	0.085	1.841	0.336	0.379	0.077	0.621	0.085	11.882	3.437
11	0.745	0.162	3.219	0.779	0.579	0.09	0.801	0.056	4.547	1.203
12	0.429	0.138	2.315	0.902	0.338	0.056	0.586	0.058	12.919	2.405
13	0.412	0.222	1.946	1.084	0.345	0.128	0.558	0.149	12.402	4.614
14	0.664	0.14	3.52	0.65	0.48	0.104	0.726	0.092	7.439	3.659

Table B.7: Mean and standard deviation of the CC, NSS, SIM, BC, and KLD metrics for each video in the ASOD60k dataset for the evaluation of the model implemented in Section 5.4.4.