SecNet: Semantic Eye Completion in Implicit Field

Anonymous Author(s)
Affiliation
Address
email

Abstract

If we take a depth image of an eye, noise artifacts and holes significantly affect the depth values on the eye due to the specularity of the sclera. This paper aims at solving this problem through semantic shape completion. We propose an end-to-end approach to train a neural network, called *SecNet* (semantic eye completion network), that predicts a point cloud with an accurate eye-geometry coupled with the semantic labels of each point. These labels correspond to the essential eye-regions, *i.e*. pupil, iris and sclera. Particularly, our work performs implicit estimation of the query points with semantic labels where both the semantic and occupancy predictions are trained in an end-to-end way. To evaluate the approach, we then use the synthetic eye-scans rendered in UnityEyes simulator environment. Compared to the state of the art, the proposed method improves the accuracy for shape-completion for 3D eye-scan by 8.2%. In practice, we also demonstrate the application of our semantic eye completion for gaze estimation.

1 Introduction

Video-oculography (VOG) has gained popularity in recent years as a method of choice for eye-tracking [35, 16, 22]. The main elements of VOG are ego-centric cameras that capture images of eye, which then undergo image-processing techniques to extract the eye movement information. 3D VOG systems on the other hand also extract torsional eye-position using iris and pupil landmarks [9]. As such, the accuracy of pupil tracking is central to the performance of VOG system, which can be significantly hampered by occlusions.

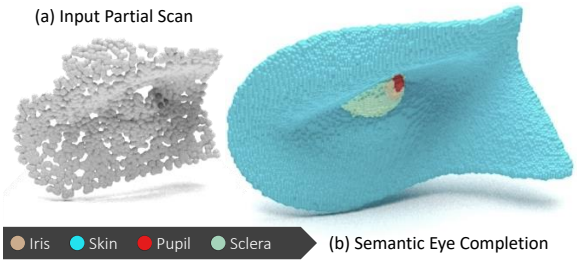


Figure 1: Given a partial scan of an eye in (a), our semantic eye completion in (b) reconstructs the surface with finer details where each point is semantically labelled.

Methods such as ellipse fitting [8], RANSAC outlier removal [12] and moving average filtering [26], and more advanced methods such as circular Hough transforms [2] for extreme pupil occlusions [27] have in particular been found useful to solve the pupil occlusion problems. However, in recent years, several algorithmic approaches that leverage 3D eye structures [20, 19] have been proposed for pupil tracking in the presence of occlusions.

Our work is focused on utilizing the 3D eye regions for pupil tracking. We leverage recent advances in 3D machine learning to reconstruct the precise 3D structure of the eye region to fill out the occluded regions. As shown in Fig. 1, shape completion is carried out on the partial scan of eye to fill out the occluded eye regions.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

Several works in recent years have addressed the problem of 3D shape completion using learning based methods. These methods can be classified based on the data-format for 3D scans. The most popular data-formats include volumetric [30, 6], meshes [10, 41], point cloud [1, 4] and implicit representation [23, 7, 3]. Among them, the implicit 3D reconstruction frameworks such as DeepSDF [23], IF-Net [3] and Points2Surf [7] provide high resolution 3D shape-completion by estimating the implicit values for random 3D query points. To estimate the implicit values, point-wise feature extractors such as PointNet features [24] and PointNet++ features [25] are commonly used.

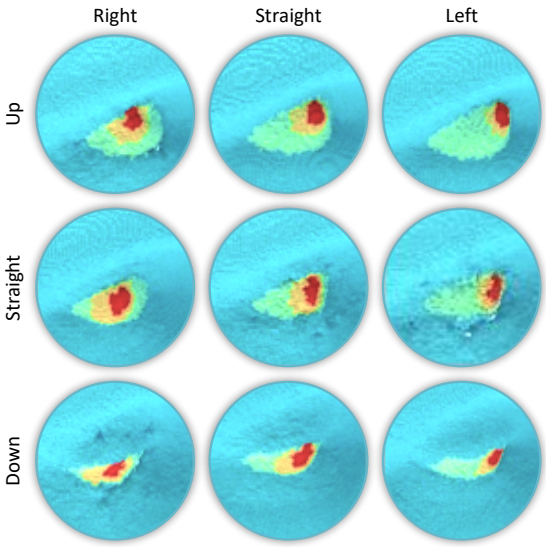


Figure 2: Nine gaze directions.

To solve the efficiency issue caused by k-nearest neighbour search in the PointNet++ feature space, this paper adopts SoftPool [40] feature as a local descriptor to construct an end-to-end model, called *SecNet*, to estimate implicit code for each given 3D query point. With the additional information of semantics used during training, our model is able to perform semantic completion in an implicit field of the eye region to precisely complete the eye surface geometries, coupled with semantics including the sclera, iris and pupil.

The training dataset for SecNet depends on the paired 2D partial scan and 3D ground truth. We create our dataset by synthesizing eye-scans using the UnityEyes [44] simulator. We then render the 3D eye-scans of 1,000 distinct people, each fixating on nine different gaze points as shown in Fig. 2.

Empirically, our proposed the semantic implicit completion model is validated on this eye region dataset, achieving state-of-the-art performance at reconstructing semantic geometries. Moreover, we empirically demonstrate that the accurate reconstruction of the completed eye region is helpful for gaze estimations.

2 Related works

This section focuses on the more general related work on 3D completion and semantic completion. In addition, since we are proposing to use the semantic eye completion for gaze estimation, we also discussed the related works that relied on depth images to estimate the gaze direction.

2.1 3D completion

There are three different ways of completing a shape from a partial scan as shown in Fig. 3: volumetric grid, point cloud and implicit surface. Early works using deep learning have relied on volumetic reconstruction because of its similarity to images, which allowed them to extend the convolution operation to 3D. For instance, 3D-EPN [5] takes TSDF volumes [43] as input and builds an encoder-decoder structure using 3D convolution. SSCNet [30] proposed to use flipped TSDF as input to perform semantic segmentation and completion at the same time. To solve the lack of 3D annotations, ForkNet [39] proposed to use the discriminator to synthetically generate new pairs of partial scan and its corresponding completed reconstruction. The main issue in such approaches is that storing 3D data in a dense volumetric grid [30, 6] consumes too much disk space and slows down inference speed for down-stream applications [6].

Point clouds were the less popular choice because of its unorganized structure. Notably, unlike volumetric data, we cannot easily apply the 3D convolution operations on them. To handle this issue, PointNet [24] proposed a solution that uses max-pooling operations to make the feature permutation invariant so that the order of the points going through the architecture does not matter. Such feature was initially proposed for 3D object classification and segmentation, which was later used in point cloud completion in FoldingNet [48], PCN [49] and AtlasNet [10].

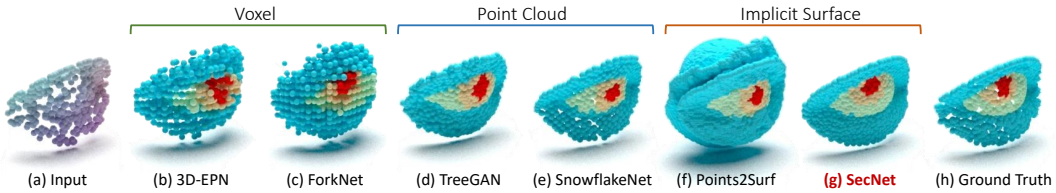


Figure 3: Given the input partial scan in (a) and the ground truth in (h), we compare different representations for semantic eye completion such as volumetirc data (b, c), point cloud (d, e) and implicit surface (f, g). Note that, except for (c) and (g) that directly infer the semantic completion, the approaches are segmented by 3D-GCN [18] to predict the semantic labels.

PointNet feature however, lacks the ability to describe the local geometry in the point cloud. This motivated the extended version PointNet++ [25] that uses k-nearest neighbor search to describe the local structure. SoftPoolNet [40] , on the other hand, is also motivated by the same objective but avoids running the time-consuming k-nearest neighbor search. Instead, this method proposes to use trainable parameters to sort the points through the feature dimension.

As we can observe in Fig. 3, completion with implicit surface generates smoother reconstruction with significantly less noise compared to volumetric and point clouds. Although their input is also based on point cloud features [7, 11], implicit 3D reconstruction such as DeepSDF [23], IF-Net [3] and Points2Surf [7] creates a fine-grained 3D shape by estimating an object surface which distinguishes the inner and outer space. Such format not only produces smoother surface reconstruction, but also reveals more local structural details compared to traditional mesh reconstruction approaches such as screened poisson reconstruction (SPR) [14].

2.2 Semantic completion

While several methods focus on completion alone [5, 23, 6, 49], there are other methods which simultaneously infer the semantic labels with the geometric completion [30, 39, 38]. For instance, SSCNet [30] uses 3D dilated convolutions to build an encoder-decoder architecture to predict semantic and occupancy value for each voxel in a predefined 3D grid. ForkNet [39] proposes a decoder with three branches which are able to generate realistic newly paired partial scan and its semantic completion to train the entire network for semantic completion.

For methods that perform completion alone [5, 40, 7], one solution to gain semantic labels is to attach a segmentation framework, *e.g*. PointNet [24] and PointCNN [17], after the geometric completion. Recently, some features are proposed for point cloud segmentation to exploit local neighbourhood such as PointNet++ [25] focusing on extracting features from local point groups. Also using the nearest neighbour search for feature extraction, KCNet [28] further aggregates the local features to investigate more complex relationships. KPConv [34] and 3D-GCN [18] make the kernel of the point cloud convolution deformable to generate better matches with different local geometries for segmentation. Although implicit reconstruction is already well investigated, an implicit reconstruction with semantic estimation is not explored. In this paper, we are proposing one of the first few works on semantic implicit completion.

2.3 Gaze estimation

There are different ways to estimate the gaze, especially when using RGB images [13]. However, in this work, we limit the scope to the utilization of 3D data which is less investigated.

To determine the direction of the user's gaze, one of the most important parameters is the position of pupil while the other is the eye's spherical center. Given the 3D structure, EMGE [50] proposes to estimate the entire spherical eyeball structure by estimating several points in pupil while RTGE [31] locates the pupil by fitting a circle to 2D scans which is back-projected to 3D for the gaze estimation. To the best of our knowledge, we are the first approach that uses semantic eye completion to perform gaze estimation.

With our solution, we encountered an issue in finding the appropriate dataset to train our models. The publicly available datasets do not provide the paried 3D partial scans and their completion. Most

datasets focus on RGB images such as SynthesEyes [45] which synthesizes 2D eye images with realistic illumination. This then prompt us to build and publish a new dataset using the UnityEyes [44] simulator engine. Similar to ShapeNet [1] for objects and ScanNet [4] for scenes, we then propose a method to build dataset based on eye meshes such as UnityEyes [44] specializing on semantic eye completion.

3 Methodology

The input to the framework is a partial scan captured by a depth camera pointing towards the eye. With N_{scan} points, each with (x, y, z) coordinate, we denote the partial scan as a point cloud $\mathcal{P}_{\text{scan}}$ which is represented as an $N_{\text{scan}} \times 3$ feature map. In practice, these scans are affected by noise and self-occlussions, *e.g.* from the eyelid and eyelashes. The objective then is to fix these issues and build a completed point cloud $\mathcal{P}_{\text{eye}}$.

Since semantic supervision is available for training, we also predict the semantic labels that includes the skin, sclera, iris and pupil. While some methods [18] utilize another inference model $\mathcal{S}$ such that the segmentation is predicted separately from the completion as $S_{\text{eye}} = \mathcal{S}(\mathcal{P}_{\text{eye}})$, we propose to use a single model to infer the semantic eye completion as $\mathcal{G}_{\mathcal{S}}(\cdot)$. This therefore estimates the geometry and the semantics at the same time.

3.1 Semantic implicit fields

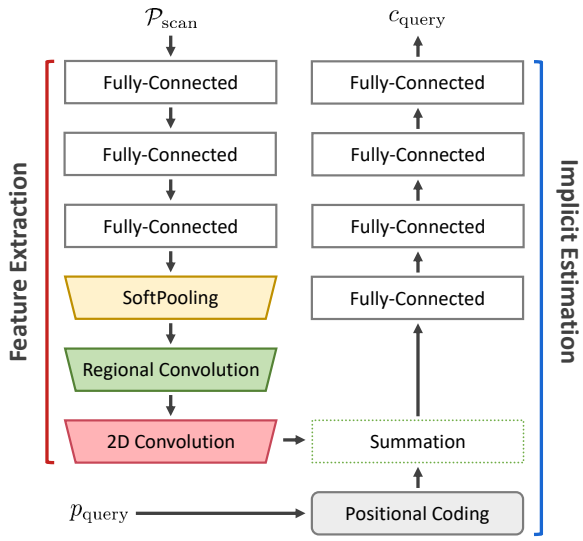
Moreover, we take advantage of the particular problem at hand. Notably, the similarities of the eye structures across different individuals and different movements allow us to effectively use implicit reconstruction. This then solves the limitation from unstructured point cloud in terms of structural accuracy and the limitation from voxel grids in terms of reconstruction resolution; consequently, leading to a reduction of noise in the reconstruction with high resolution. This is validated in Fig. 3. In addition, we noticed that our method even produced a denser and smoother reconstruction than the ground truth.

Inspired by the works of IF-Net [3] and Points2Surf [7], we also learn implicit values between a set of query points and the mesh surface. For each query point, these methods predict a value between -1 to 1, where a query point on the surface is at the zero-crossing. The difference between traditional implicit surface learning and our model is that we propose to present the semantic labels in addition to the geometry, which we call *semantic implicit field* (SIF). Therefore, in our work, given an arbitrary query point p_{query}, we can simplify the framework to a classification task where the architecture predicts if the point is an *empty* space, or part of the *skin*, *sclera*, *iris* or *pupil*. We refer the classification result as the *semantic code* c_{query} of the query point such that $c_{\text{query}} = \mathcal{G}_{\mathcal{S}}(\mathcal{P}_{\text{scan}}, p_{\text{query}})$. This implies that the output eye reconstruction $\mathcal{P}_{\text{eye}}$ is presented by all the query points that are not empty. In practice, assuming that the partial scan is normalized to a unit cube, we sample the query points randomly around the partial scan within a Chamfer distance of 0.3.

3.2 SecNet architecture

The architecture for $\mathcal{G}_{\mathcal{S}}(\cdot)$ is summarized in Table 1, where we build an encoder-decoder structure. Here, the encoder processes the partial scan $\mathcal{P}_{\text{scan}}$ and produces the latent feature that describes the global structure. Having the latent feature and a query point p_{query}, the decoder runs the implicit estimation that finds the semantic code c_{query} which classifies whether the point is empty or the specific part of the eye.

In particular, the encoder first randomly sub-samples the $\mathcal{P}_{\text{scan}}$ into $N_{\text{in}} = 2048$ points to have a constant tensor as input. These points are fed to a 3-layer MLP that generates an output dimension of 8. We then use the SoftPool [40] operation with the number of regions $N_f = 8$ and the number of regional points $N_r = 32$, which produces an 8-region feature map with the shape of $[256, 8]$. This is processed by a regional convolution [40] with a kernel size $D_{\text{kernel}} = [N_p, N_f]$ for all 8 regions, which covers $N_p = 8$ points with zero padding in each region. We then add a 2D convolution with kernel size $D_{\text{kernel}} = [N_r \times N_f, N_f]$, resulting in a 64-dimensional vector as the latent feature. Since the encoder is only dependent on the partial scan, the latent feature is constant across all the query points in the decoder.



Modules	Parameters
MLP (encoder)	$D_{\text{out}} = [512, 512, 8]$
Soft Pooling	N_r = 32, N_f = 8
Regional Conv.	N_p = 8, $D_{\text{out}} = 8$, $D_{\text{kernel}} = [8, 8]$
2D Conv.	D_{out} = 64, $D_{\text{kernel}} = [256, 8]$
Positional Coding	$D_{\text{out}} = 64$
MLP (decoder)	$D_{\text{out}} = [16, 32, 64, 5]$

Table 1: Architecture of SecNet with the corresponding hyperparameters for each module where D_{out} represents the output dimensions.

Every query point in the decoder is converted to a positional code using SIREN [29], having the same dimension as the latent feature. Thereafter, the sum of the positional coding and the latent feature serves as the input to the 4-layer MLP to estimate the final semantic code c_{query} with a softmax activation.

We summarize the numerical values of our architecture in Table 1. This table shows the architecture on top and the corresponding parameters for each layer at the bottom.

To train the proposed encoder-decoder model, we impose the per-category binary cross entropy $\epsilon_c(\cdot, \cdot)$ such that

$$\mathcal{L}_{\text{semantic}} = \sum_c \epsilon_c(c_{\text{query}}, c_{\text{gt}}) = \sum_c \epsilon_c\left(\mathcal{G}_{\text{S}}(\mathcal{P}_{\text{scan}}, p_{\text{query}}), c_{\text{gt}}\right) \tag{1}$$

sums up the entropy for all categories. Given this loss function, we train the model $\mathcal{G}_{\mathcal{S}}$ with a batch size of 64. We employ the Adam optimizer [15] with a learning rate of 0.0001 while the exponential decay rates β_1 and β_2 are set to 0.9 and 0.999, respectively.

3.3 Gaze estimation

As a by-product of our semantic eye completion, we estimate the gaze direction through the semantic points. Similar to RTGE [31], we solve this problem by estimating a 3D vector from the center of the eyeball to the center of the pupil. To find the centers, we use all the points on the sclera to fit a sphere that represents the eyeball; then, take the average point of all the points on the iris. The gaze direction is finally estimated as the vector that connects the center of the sphere and the average point. For the sphere, we use an energy optimization to estimate its center $p_{\text{center}} = (x_c, y_c, z_c)$ as well as

its radius r. This minimizes the loss

$$\mathcal{L}_{\text{eyeball}} = \sum_{i}^{N_{\text{sclera}}} \left| \left\| p^{i}_{\text{sclera}} - p_{\text{center}} \right\|^{2} - r^{2} \right| , \tag{2}$$

summing up the absolute errors from all the N_{sclera} points labelled as sclera p^{i}_{sclera} in the semantic eye completion.

4 Dataset

We generate the dataset by rendering pairs of partial scans and their corresponding semantic completion using the UnityEyes [44] mesh models. To generalize for the gaze estimation, we rotate the eyeball towards nine gaze directions as shown in Fig. 2, including up-right, up, up-left, right, straight, left, down-right, down and down-left. With 1,000 identities from UnityEyes, we then have a total of 9,000 pairs in the dataset. For the experiments, we split our dataset with 800 identities for training and 200 for testing. One of the main advantages in having depth images or partial scans as input is the privacy preservation in training or during inference.

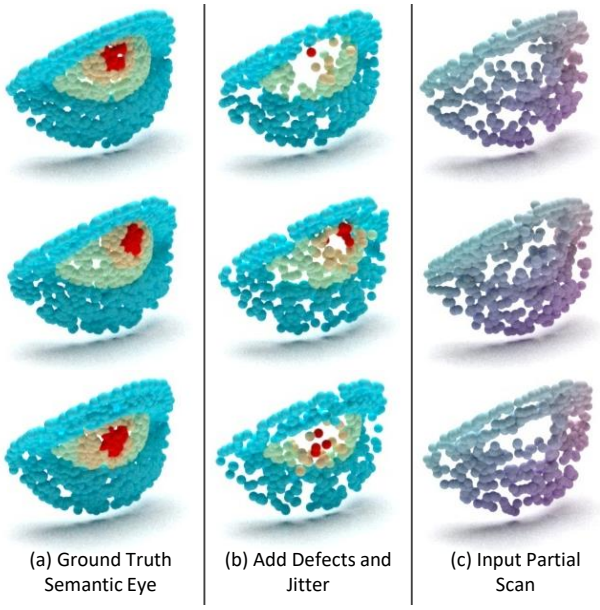


Figure 4: Examples of our dataset.

Fig. 4 shows some examples of the process that the model undergo when creating the dataset. Since the dataset is synthetically rendered, we impose the defects and self-occlusion by randomly dropping 61.2% points for pupil, 74.9% of iris, 29.7% of sclera and 9.0% of skin on every mesh models as shown in Fig. 4(b). In addition, the surface also incorporates jitter in order to mimic the sensor noise. The jitter is defined by a Gaussian distribution with zero mean and a 0.05 standard deviation for points on the sclera, iris and pupil. For this dataset, Fig. 4(c) illustrates the example input partial scans that we use for this evaluation. Noticeably, without visualizing the ground truth semantic labels in Fig. 4(c), identifying the regions on the eye or the gaze direction from the three images becomes very difficult.

5 Experiments

To highlight our contributions in this work, we conduct the following 3 experiments: synthetic eye augmentation, semantic eye completion and gaze estimation. We empirically demonstrate the advantages of our approaches using the dataset from Sec. 4.

5.1 Semantic eye completion

We evaluate the semantic completion on the eye region on the dataset captured from UnityEyes [44] models. The point clouds of the partial scans serves as the input for both training and testing. Their corresponding ground truth completed shape is presented in terms of a point cloud with semantic labels. Our evaluations are carried out across 4 categories including skin, sclera, iris and pupil. Note that, across all methods, the input partial scan is normalized into the same scale of point coordinates ranging between -0.5 to 0.5.

This evaluation compares against the state-of-the-art completion methods that reconstruct using volumetric data, point cloud or implicit surface. While Fig. 3 shows the qualitative results of different

	Method	skin	sclera	iris	pupil	*avg.*
Voxel	3D-EPN [5]	13.43	22.09	19.43	15.96	17.73
	ForkNet [39]	17.04	14.75	18.16	14.78	16.18
Point Cloud	PointNet++ [25]	9.72	10.24	12.73	11.85	11.13
	FoldingNet [48]	9.35	10.23	12.29	11.68	10.89
	TopNet [33]	8.82	10.22	11.82	11.02	10.48
	AtlasNet [10]	8.15	9.70	11.18	10.64	9.92
	PCN [49]	7.48	9.69	11.12	10.29	9.65
	MSN [21]	7.01	9.10	10.63	9.77	9.13
	SoftPoolNet [40]	6.54	8.78	9.79	9.46	8.65
	GRNet [47]	6.22	8.70	9.68	9.14	8.44
	PMP-Net [42]	5.74	8.26	8.98	8.83	7.96
	CRN [37]	5.57	8.23	8.98	8.81	7.90
	SnowflakeNet [46]	4.93	7.48	8.76	8.68	7.46
Implicit	IF-Net [3]	5.43	7.98	7.95	7.09	7.12
	Points2Surf [7]	4.93	7.45	7.48	6.91	6.70
	Ours	**4.21**	**6.99**	**7.17**	**6.25**	**6.15**

Table 2: Evaluation of the semantic eye completion. We measure the Chamfer distance for each category; and, compute the average across all categories.

approaches, Table 2 highlights the numerical comparison between them. This table shows that we achieve the state-of-the-art results, reaching an average L1-Chamfer distance of 6.15×10^{-3} on all the categories. It also shows that we have the best results across all the four categories. It is noteworthy to mention that only ForkNet [39] and our approach perform semantic completion, while other methods can only infer the geometric completion. Due to this, for the other methods, we apply 3D-GCN [18] on the reconstruction to find the semantic labels.

If we investigate closely on the comparison against the volumetric methods, their error are significantly higher, *e.g*. ForkNet [39] has an average Chamfer distance of 16.18×10^{-3}, since they only use a grid with an output dimension of $[64, 64, 64]$. To evaluate in the same metric, we convert the grid into a point cloud before evaluating the Chamfer distance. From Fig. 3, we can observe that they have an obvious disadvantage due to their low resolution.

As for point cloud approaches, we compare our semantic eye completion results with some recently proposed methods such as FoldingNet [48], PCN [49], MSN [21], GRNet [47] and SoftPoolNet [40]. All these approaches reconstructs a point cloud which is further re-sampled to 4,096 points for evaluation. Compared to SnowflakeNet [46] which is the state-of-the-art point cloud completion approach, our method achieves decreased the Chamfer distance by 1.31×10^{-3}.

Lastly, when we compare against other implicit method, Points2Surf [7] also performs well with a Chamfer distance of 6.70×10^{-3}. However, our proposed architecture performs the best among all listed approaches in Table 2 with an error of 6.15×10^{-3}.

5.2 Gaze estimation

Since we can convert the semantic eye completion to gaze direction through Sec. 3.3, this section focuses on the evaluation of the gaze direction. In addition to the gaze from semantic completion, we also include the related work that designed for gaze estimation such as EMGE [50] and RTGE [31]. These methods directly locate the pupil and estimate the center of eyeball from 2D partial scan without eye completion. We also include 3D-GCN [18] which segments the input partial scan into parts prior to the gaze estimation.

We first consider this as a classification problem where we match the estimated gaze based on the nine directions. Comparing with other methods in Table 3, our approach reached a classification accuracy of 97.6% which is significantly higher than any other approach.

		Method	*Accuracy*	*Cosine Distance*	*Model Size (MB)*	*Time (seconds)*
Direct Gaze Estimation		EMGE [50]	42.1%	0.637	–	0.14
		RTGE [31]	56.9%	0.691	–	0.08
		3D-GCN [18]	61.8%	0.745	6.6	0.82
Gaze from Semantic Eye Completion	*Voxel*	3D-EPN [5]	81.4%	0.802	420.0	0.82
		ForkNet [39]	83.8%	0.809	362.0	1.12
	Point Cloud	PointNet++ [25]	82.9%	0.781	29.7	2.33
		FoldingNet [48]	84.6%	0.807	19.2	0.05
		TopNet [33]	85.1%	0.822	79.9	0.61
		AtlasNet [10]	85.2%	0.821	2.0	0.32
		PCN [49]	87.3%	0.823	54.8	0.11
		MSN [21]	88.0%	0.830	12.0	0.21
		SoftPoolNet [40]	89.2%	0.842	37.2	0.04
		GRNet [47]	91.6%	0.857	293.0	0.88
		PointCNN [17]	87.6%	0.826	497.0	1.20
		PMP-Net [42]	90.6%	0.850	22.0	4.21
		CRN [37]	93.1%	0.884	61.5	2.73
	Implicit	IF-Net [3]	93.6%	0.909	29.4	9.27
		Points2Surf [7]	94.3%	0.921	24.0	12.64
		Ours	**97.6%**	**0.971**	9.7	0.19

Table 3: Evaluation of the gaze direction classification and estimation with the corresponding model size and inference time. The table is divided into two regions. The methods on top directly use the depth image to find the gaze; while, the methods at the bottom estimates the gaze based on the semantic eye completion. Note that [50, 31] does not depend on a parameterized inference model.

Instead of relying only on discrete values, we also considered the cosine distance to evaluate the estimated gaze from the ground truth, which is the dot product of the two vectors. Here, our approach also produces the best performance of 0.971.

5.3 Efficiency

We also evaluate the processing time and the corresponding memory footprint of each model, which is summarized in Table 3. This table illustrates that our inference time at 0.19 seconds is much faster than the other implicit reconstruction methods such as DeepSDF [23] at 9.72 seconds and Points2Surf [7] at 12.64 seconds. This is because the other methods require zero-crossing in reconstruction while our method does not. In addition, our point-wise implicit estimation is conditioned on a SoftPool feature from the encoder, which is processed once for each partial scan. Points2Surf, on the other hand, decodes the implicit values using QSTN [11] which depends on analyzing the local point cloud patches. This implies that it needs to be executed repetitively for each query point. As a consequence, we can reduce the time by focusing on areas surrounding the partial scan to decrease the number of query points to process. Overall, we do not attain the lowest memory footprint or the lowest inference time. However, we argue that our approach has a good trade-off between the two parameters.

5.4 Ablation study

We perform an ablation study to understand the effect of changes in the hyperparameters. Table 4 summarizes this study. We noticed that the MLP in the encoder does not significantly change the completion performance as long as the output dimension of the first layer of the MLP is larger than 256. Having values above 512 only improves the average Chamfer distance from 6.15×10^{-3} to 6.09×10^{-3}.

Since our latent feature is extracted by SoftPool [40] operators, we validate the changes in performance by adapting different input feature dimension N_f and number of points N_r chosen from each of the sequential feature map. Our experimental results show that the performance ranges from 6.10×10^{-3}

Modules	Parameters	Chamfer Distance
positional coding	Gaussian [32]	6.42
	SIREN [29]	**6.15**
	sinusoidal [36]	6.31
MLP (encoder)	$D_{\text{out}} = [512, 256, 8]$	6.99
	$D_{\text{out}} = [256, 512, 8]$	7.33
	$D_{\text{out}} = [512, 512, 8]$	**6.15**
	$D_{\text{out}} = [1024, 512, 8]$	6.12
	$D_{\text{out}} = [512, 1024, 8]$	6.09
Softpool [40]	N_r = 16, N_f = 8	6.84
	N_r = 32, N_f = 4	8.07
	N_r = 32, N_f = 8	**6.15**
	N_r = 32, N_f = 16	6.10
	N_r = 64, N_f = 8	6.14
MLP (decoder)	$D_{\text{out}} = [16, 16, 64, 5]$	7.04
	$D_{\text{out}} = [16, 32, 32, 5]$	6.60
	$D_{\text{out}} = [16, 32, 64, 5]$	**6.15**
	$D_{\text{out}} = [16, 64, 64, 5]$	6.13
	$D_{\text{out}} = [16, 32, 128, 5]$	6.11

Table 4: Ablation study on network hyperparameters. The results in bold indicate the chosen parameters in architectural design which balance the accuracy and model size.

to 6.84×10^{-3}. This is validated by [40] where we can reach a good performance as long as the feature dimension N_f is larger than 4. For the MLP in decoder, we found that performance saturates when $D_{\text{out}} = [16, 32, 64, 5]$.

6 Conclusion

In this paper, we propose to complete the eye region through *semantic implicit field*. Using our semantic eye completion, we also introduce a more practical use-case, *i.e*. gaze estimation. We achieve the state-of-the-art performance for both semantic eye completion and gaze estimation. Since we propose a new problem in semantic completion and a new type of solution for gaze estimation, we propose a simple way to build the dataset for semantic completion eyes based on UnityEyes [44] meshes to train and evaluate the models.

References

[1] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[2] Noureddine Cherabit, Fatma Zohra Chelali, and Amar Djeradi. Circular hough transform for iris localization. *Science and Technology*, 2(5):114–121, 2012.

[3] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3d shape reconstruction and completion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2020.

[4] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas A Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 2, page 10, 2017.

[5] Angela Dai, Charles Ruizhongtai Qi, and Matthias Nießner. Shape completion using 3d-encoder-predictor cnns and shape synthesis. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 3, 2017.

[6] Angela Dai, Daniel Ritchie, Martin Bokeloh, Scott Reed, Jürgen Sturm, and Matthias Nießner. Scancomplete: Large-scale scene completion and semantic segmentation for 3d scans. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 1, 2018.

[7] Philipp Erler, Paul Guerrero, Stefan Ohrhallinger, Niloy J Mitra, and Michael Wimmer. Points2surf learning implicit surfaces from point clouds. In *European Conference on Computer Vision*, pages 108–124. Springer, 2020.

[8] Andrew Fitzgibbon, Maurizio Pilu, and Robert B Fisher. Direct least square fitting of ellipses. *IEEE Transactions on pattern analysis and machine intelligence*, 21(5):476–480, 1999.

[9] Sonia Goni, Javier Echeto, Arantxa Villanueva, and Rafael Cabeza. Robust algorithm for pupil-glint vector detection in a video-oculography eyetracking system. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 4, pages 941–944. IEEE, 2004.

[10] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan C. Russell, and Mathieu Aubry. A papier-mâché approach to learning 3d surface generation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[11] Paul Guerrero, Yanir Kleiman, Maks Ovsjanikov, and Niloy J Mitra. Pcpnet learning local shape properties from raw point clouds. In *Computer Graphics Forum*, volume 37, pages 75–85. Wiley Online Library, 2018.

[12] Yong-Dian Jian and Chu-Song Chen. Two-view motion segmentation with model selection and outlier removal by ransac-enhanced dirichlet process mixture models. *International Journal of Computer Vision*, 88(3):489–501, 2010.

[13] Li Jianfeng and Li Shigang. Eye-model-based gaze estimation by rgb-d camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 592–596, 2014.

[14] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)*, 32(3):1–13, 2013.

[15] Diederick P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

[16] Agostina J Larrazabal, CE García Cena, and César Ernesto Martínez. Video-oculography eye tracking towards clinical applications: A review. *Computers in biology and medicine*, 108:57–66, 2019.

[17] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In *Advances in Neural Information Processing Systems*, pages 820–830, 2018.

[18] Zhi-Hao Lin, Sheng-Yu Huang, and Yu-Chiang Frank Wang. Convolution in the cloud: Learning deformable kernels in 3d graph convolution networks for point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1800–1809, 2020.

[19] Meng Liu, You Fu Li, and Hai Liu. 3d gaze estimation for head-mounted devices based on visual saliency. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10611–10616. IEEE, 2020.

[20] Meng Liu, Youfu Li, and Hai Liu. Robust 3-d gaze estimation via data optimization and saliency aggregation for mobile eye-tracking systems. *IEEE Transactions on Instrumentation and Measurement*, 70:1–10, 2021.

[21] Minghua Liu, Lu Sheng, Sheng Yang, Jing Shao, and Shi-Min Hu. Morphing and sampling network for dense point cloud completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11596–11603, 2020.

[22] Nitinraj Nair, Aayush Kumar Chaudhary, Rakshit Sunil Kothari, Gabriel Jacob Diaz, Jeff B Pelz, and Reynold Bailey. Rit-eyes: realistically rendered eye images for eye-tracking applications. In *ACM Symposium on Eye Tracking Research and Applications*, pages 1–3, 2020.

[23] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019.

[24] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017.

[25] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[26] Thoriq Satriya, Sunu Wibirama, and Igi Ardiyanto. Robust pupil tracking algorithm based on ellipse fitting. In *2016 International Symposium on Electronics and Smart Devices (ISESD)*, pages 253–257, 2016.

[27] Muhammad Taufiq Setiawan, Sunu Wibirama, and Noor Akhmad Setiawan. Robust pupil localization algorithm based on circular hough transform for extreme pupil occlusion. In *2018 4th International Conference on Science and Technology (ICST)*, pages 1–5, 2018.

[28] Yiru Shen, Chen Feng, Yaoqing Yang, and Dong Tian. Mining point cloud local structures by kernel correlation and graph pooling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4548–4557, 2018.

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

[30] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017.

[31] Li Sun, Zicheng Liu, and Ming-Ting Sun. Real time gaze estimation with a consumer depth camera. *Information Sciences*, 320:346–360, 2015.

[32] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 33:7537–7547, 2020.

[33] Lyne P Tchapmi, Vineet Kosaraju, Hamid Rezatofighi, Ian Reid, and Silvio Savarese. Topnet: Structural point cloud decoder. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 383–392, 2019.

[34] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6411–6420, 2019.

[35] Josef N van der Geest and Maarten A Frens. Recording eye movements with video-oculography and scleral search coils: a direct comparison of two methods. *Journal of neuroscience methods*, 114(2):185–195, 2002.

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[37] Xiaogang Wang, Marcelo H. Ang Jr. , and Gim Hee Lee. Cascaded refinement network for point cloud completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[38] Y. Wang, D. J. Tan, N. Navab, and F. Tombari. Adversarial semantic scene completion from a single depth image. In *2018 International Conference on 3D Vision (3DV)*, 2018.

[39] Yida Wang, David Joseph Tan, Nassir Navab, and Federico Tombari. Forknet: Multi-branch volumetric semantic completion from a single depth image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8608–8617, 2019.

[40] Yida Wang, David Joseph Tan, Nassir Navab, and Federico Tombari. Softpoolnet: Shape descriptor for point cloud completion and classification. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 70–85, Cham, 2020. Springer International Publishing.

[41] Xingkui Wei, Zhengqing Chen, Yanwei Fu, Zhaopeng Cui, and Yinda Zhang. Deep hybrid self-prior for full 3d mesh generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5805–5814, 2021.

[42] Xin Wen, Peng Xiang, Zhizhong Han, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Yu-Shen Liu. Pmp-net: Point cloud completion by learning multi-step point moving paths. *arXiv preprint arXiv:2012.03408*, 2020.

[43] Diana Werner, Ayoub Al-Hamadi, and Philipp Werner. Truncated signed distance function: experiments on voxel size. In *International Conference Image Analysis and Recognition*, pages 357–364. Springer, 2014.

[44] Erroll Wood, Tadas Baltrušaitis, Louis-Philippe Morency, Peter Robinson, and Andreas Bulling. Learning an appearance-based gaze estimator from one million synthesised images. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, pages 131–138, 2016.

[45] Erroll Wood, Tadas Baltrusaitis, Xucong Zhang, Yusuke Sugano, Peter Robinson, and Andreas Bulling. Rendering of eyes for eye-shape registration and gaze estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3756–3764, 2015.

[46] Peng Xiang, Xin Wen, Yu-Shen Liu, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Zhizhong Han. Snowflakenet: Point cloud completion by snowflake point deconvolution with skip-transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5499–5509, 2021.

[47] Haozhe Xie, Hongxun Yao, Shangchen Zhou, Jiageng Mao, Shengping Zhang, and Wenxiu Sun. Grnet: Gridding residual network for dense point cloud completion. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 365–381, Cham, 2020. Springer International Publishing.

[48] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 206–215, 2018.

[49] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. Pcn: Point completion network. In *2018 International Conference on 3D Vision (3DV)*, pages 728–737. IEEE, 2018.

[50] Xiaolong Zhou, Haibin Cai, Zhanpeng Shao, Hui Yu, and Honghai Liu. 3d eye model-based gaze estimation from a depth sensor. In *2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 369–374. IEEE, 2016.