IEEE Robotics and Automation Letters (RAL) paper presented at the
2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

SilhoNet-Fisheye: Adaptation of A ROI Based Object Pose Estimation Network to Monocular Fisheye Images

Gideon Billings[1] and Matthew Johnson-Roberson[1]

***Abstract*— There has been much recent interest in deep learning methods for monocular image based object pose estimation. While object pose estimation is an important problem for autonomous robot interaction with the physical world, and the application space for monocular-based methods is expansive, there has been little work on applying these methods with fisheye imaging systems. Also, little exists in the way of annotated fisheye image datasets on which these methods can be developed and tested. The research landscape is even more sparse for object detection methods applied in the underwater domain, fisheye image based or otherwise. In this work, we present a novel framework for adapting a ROI-based 6D object pose estimation method to work on full fisheye images. The method incorporates the gnomic projection of regions of interest from an intermediate spherical image representation to correct for the fisheye distortions. Further, we contribute a fisheye image dataset, called UWHandles, collected in natural underwater environments, with 6D object pose and 2D bounding box annotations.**

I. Introduction

The advantages of fisheye imaging systems in robotics applications has long been recognized. With technological improvements in imaging sensor resolution and dynamic range, fisheye cameras can capture significantly greater information about the surrounding environment without appreciably increasing the imaging sensor footprint, compared to their perspective model counterparts. However, little work has been done on applying CNN based methods to the problem of 6D object pose estimation on full fisheye images. Dealing with fisheye images is challenging, due to the large distortions and viewpoint ambiguities arising from the wide field of view. We address the problem of 6D object pose estimation in full fisheye images by proposing a method whereby the image is first projected to the surface of a sphere, where we mathematically define a consistent apparent viewpoint which the network is trained to predict. The true orientation relative to the fisheye frame can then be recovered using the predicted translation. The gnomonic projection is used in our method to undistort the region of interest (ROI) from the sphere surface, and we investigate applying this projection both before and after the feature extraction stage.

Further, while deep learning has greatly advanced the state of-the-art in terrestrial based visual methods for object detection and pose estimation, the challenges of collecting underwater datasets for training these methods and the limited access to underwater environments has hindered progress in leveraging these advancements for underwater applications. We address this challenge by introducing an easy and efficient method for collecting fiducially grounded monocular images of underwater scenes containing known objects, and we contribute a tool for annotating the 6D object poses and bounding boxes in the image sequences. We also contribute an underwater visual dataset of three different graspable handles with annotated ground truth poses and bounding boxes. The dataset was collected with a fisheye camera mounted on the wrist of an ROV manipulator, with objects appearing in different arrangements in diverse natural seafloor environments.

In summery, we present the following contributions: 1) A framework for adapting ROI-based networks for predicting 6D object pose from monocular images to work on full fisheye images, through an intermediate mapping onto a sphere and ROI processing through the gnomonic projection. This adaptation is demonstrated with the SilhoNet method presented in [1]; 2) UWHandles[2], an underwater monocular fisheye image dataset of handle objects, with annotated 6D pose and 2D bounding boxes, collected in natural seafloor environments. We also release the annotation tool, VisPose[3], used to process the dataset;

The rest of this paper is organized in the following sections: Section II discusses related work; Section III presents our methods for adapting SilhoNet to the fisheye domain; Section IV presents the UWHandles dataset; Section V presents the experimental results; and Section VI concludes the paper.

II. Related Work

In general, state-of-the-art works that apply CNN methods to full fisheye images process the raw images directly through the network without special consideration of the fisheye distortions [2]–[4]. These networks are mostly applied to the problems of segmentation or ROI detection in the fisheye images. Due to the sparsity of available benchmarking datasets for fisheye images, these works report their results on synthetic datasets, generated by projecting perspective images to distorted fisheye images. Zhu *et al.* [5] used a CNN in the prediction of ground vehicle positions relative to an aerial fisheye imaging platform. They directly train the CNN on the raw fisheye images to generate ROI proposals. They assume the detected object is on the ground plane and fuse measurements from height and orientation sensors on the

This work was supported by the NASA award NNX16AL08G.

[1]Gideon Billings and Matthew Johnson-Roberson are with Department of Naval Architecture and Marine Engineering, University of Michigan, 2600 Draper Dr. Ann Arbor, MI 48109, USA gidobot@umich.edu

[2]https://github.com/gidobot/UWHandles

[3]https://github.com/gidobot/VisPose

camera platform to recover only the object's 3D translation in the world. Salem *et al.* [6] extended the Cascaded Pose Regression algorithm to estimate the 3D pose of mice in fisheye images from detected 3D keypoints. However, their method incorporates priors about the structured lab environment, and the fisheye camera is fixed in the scene, allowing them to easily segment the mice from the background image. In contrast to these works, our method incorporates knowledge of the fisheye distortion model through a spherical mapping, which improves network performance and is also necessary to create visually consistent pose annotations which can be regressed directly from ROI proposals across the full fisheye field of view. Further, we report the performance of our method on a real fisheye dataset captured in a natural unstructured environment.

Closely related to fisheye image processing is the extensive body of work on omni-directional imaging, as both fisheye and omni-directional image distortions can be represented on a sphere. Beyond naively applying CNNs directly to a flattened equirectangular projection of an omni-directional image, which has been shown to suffer from the nonlinear distortions of the spherical mapping to the plane and attain sub-optimal performance [7], the methods of dealing with omni-directional distortions can be roughly categorized under three approaches: generating multiple perspective projections from the sphere, such as cube map, and processing each projection separately through the CNN [8]; adapting the kernel sampling locations based on a spherical distortion model or a learned mapping [9]–[12]; re-sampling the spherical image based on a uniform sampling geometry such as the icosahedron, and processing the spherical representation with specialized convolution operations [13]–[16]; or transforming the spherical feature signals and convolution operations into the spectral domain, typically by representation of the spherical image as a graph [17]–[19]. Methods that operate on multiple perspective projections suffer from discontinuities at the projection borders, due to variance in feature appearance on different tangent plane mappings. Methods that operate on graphical representations of the sphere in the spectral domain are memory limited in scaling to full resolution images and have some level of rotation invariance in the convolution response function, which is undesirable when regressing 6D object pose. Methods that re-sample the convolution kernel sampling location based on a learned or distortion based mapping are most relevant to our work. The methods of Coors *et al.* [10] and Zhao *et al.* [9] sample regular kernel locations on a tangent plane and then project the sampling locations to the spherical surface, encoding the spherical distortions directly into the convolution operation. Zhao *et al.* [20] adapts a region proposal network with the distortion aware convolutions of [9], [10] in a two-stage architecture to predict region proposals from omni-directional images. However, these distortion aware convolutions are designed to operate on full 360°images. Because fisheye images represent only a partial view of the sphere, they can also be analyzed under different planar projections than omni-directional images. Further, application of omni-directional CNNs to 6D object pose estimation is so far lacking in the literature. Our method takes inspiration from these prior works [9], [10] that incorporate a mapping to a spherical surface and the Gnomonic projection to a tangent plane to deal with feature distortions in omni-directional images. The main technical contribution of our work is the mathematical formulation of applying a spherical mapping and the Gnomonic projection to the problem of 6D object pose estimation in wide field-of-view imagery. Though we develop the method assuming the equidistant fisheye projection model, the formulation is valid for any camera projection that can be mapped to a spherical surface, including omni-directional images.

The body of work applying CNN methods to underwater imagery is mostly limited to the problems of species detection and classification [21]–[29], or underwater image correction [30], [31] on perspective images. Kuang *et al.* [32] used a simple color distortion model based on image depth to generate a synthetic dataset of omni-directional images that were color cast as though captured underwater. They trained a distortion aware CNN to predict image depth from an omni-directional image, and reported results on their synthetic dataset. While they did not test with real omni-directional data, the perspective image equivalent of their method performed very poorly on real underwater images. Most related to our work in the underwater domain is the work of Jeon *et al.* [33], who proposed a CNN based method for underwater object detection and pose estimation, using a synthetic dataset generated from CAD models to train the network. However, the objects used in their dataset were very simple, and their tests were limited to tank environment with high contrast between the object models and the scene background. Further, they only regressed the 3D orientation of the detected objects. Also related to our work is Nielsen *et al.* [34], where a PoseNet CNN was trained to regress the 6D pose of a mock-up sub-sea connector relative to a small ROV. The dataset was collected in a tank environment, with high contrast between the connector target and the low featured background. In contrast to these works, our method addresses the problem of full 6D object pose estimation from monocular underwater images captured in wild unstructured environments. Further, our method is applied to full view fisheye images, which capture a significantly greater field of view over perspective images. Also, our dataset is composed with visually challenging handle objects used to manipulate ROV tools in real life applications.

Largely, works targeting deep learning methods applied to omni-directional or fisheye images resort to generating or collecting their own custom datasets, due to a lack of large scale annotated image datasets of this type. Many works simply project annotated perspective images to the distorted omni-directional or fisheye domain, as a simple way to generate a proxy dataset. Some recent work seeks to address this lack of annotated fisheye datasets [35], [36]. However, no such annotated dataset exists for fisheye images in the underwater domain. We address this problem by releasing our annotated UWHandles dataset, along with the method

and annotation tool used to collect and process the images.

III. Method

Special care must be taken in regressing 6D pose from full fisheye images, as there can be large distortions and ambiguity in the object viewpoint (Fig. 2). In the following sections, we outline how we use an intermediate spherical representation and the gnomonic projection to attain visually consistent pose annotations, followed by an overview of three different adaptions of the SilhoNet method[1] for 6D pose prediction from full fisheye images (Fig.1).

A. Spherical Mapping and Gnomonic Projection

While a class of different projection models exist for fisheye cameras [37], the model followed by the camera system used in this work, and the most common model in practice, is the equidistant projection

$$R = f\theta \tag{1}$$

where θ is the angle in radians from a point in the world to the optical axis, f is the lens focal length, and R is the radial position of the point projected on the imaging plane. A major challenge of fisheye images when regressing the object orientation is the large space of visual ambiguity. We define the global reference frame as coincident with the fisheye camera frame. As the angle between the object center in the world to the camera optical axis increases, there is increasing discrepancy between the object orientation relative to the global frame and the apparent orientation relative to a cropped ROI (Fig. 2). We deal with this visual ambiguity by first mapping the fisheye image onto the unit sphere. The mapping between the pixel coordinates (x, y) on the fisheye image with focal length f and the polar coordinates (θ, ϕ) on the unit sphere is given as

$$r = \sqrt{x^2 + y^2};\ \rho = r/f;\ z = \frac{r}{\tan\rho} \tag{2}$$

$$\theta = \sin^{-1}\left(\frac{y}{\sqrt{x^2 + y^2 + z^2}}\right);\ \phi = \tan^{-1}\frac{x}{z}. \tag{3}$$

The inverse mapping can also be calculated by first converting the spherical coordinates to cartesian and then projecting onto the image plane with the fisheye model

$$x_s = \cos\theta\sin\phi;\ y_s = \sin\theta;\ z_s = \cos\theta\cos\phi \tag{4}$$

$$\rho = \cos^{-1}\left(\frac{z_s}{\sqrt{x_s^2 + y_s^2 + z_s^2}}\right);\ r = f\rho \tag{5}$$

$$x = \frac{x_s r}{\sqrt{x_s^2 + y_s^2}};\ y = \frac{y_s r}{\sqrt{x_s^2 + y_s^2}}. \tag{6}$$

By mapping the fisheye image to a unit sphere centered on the global origin, we can define the apparent viewpoint of the object as the appearance of the object when projected onto a tangent plane centered on the vector extending from the sphere center to the center of the object. The projection from the sphere onto the tangent plane is known as a gnomonic projection and has a long history in mapping as well as recent application in omni-directional CNN methods [9], [10]. Given a spherical mapping of an image and the tangent plane centered on the sphere at polar coordinates (θ_0, ϕ_0), the gnomonic projection of the spherical point (θ,ϕ) onto the tangent plane is given as

$$x = \frac{\cos\theta\sin(\phi - \phi_0)}{\sin\theta_0\sin\theta + \cos\theta_0\cos\theta\cos(\phi - \phi_0)} \tag{7}$$

$$y = \frac{\cos\theta_0\sin\theta - \sin\theta_0\cos\theta\cos(\phi - \phi_0)}{\sin\theta_0\sin\theta + \cos\theta_0\cos\theta\cos(\phi - \phi_0)}, \tag{8}$$

and an optimized inverse mapping from the tangent plane onto the sphere is given as

$$\theta = \sin^{-1}\left(\frac{\sin\theta_0 + y\cos\theta_0}{\sqrt{1 + x^2 + y^2}}\right) \tag{9}$$

$$\phi = \phi_0 + \tan^{-1}\left(\frac{x}{\cos\theta_0 - y\sin\theta_0}\right), \tag{10}$$

where x and y are the coordinates of the pixel on the tangent plane normalized by the virtual perspective camera focal length f_p [38], [39]. The gnomonic projection is core to our method of regressing the object 6D pose from ROI proposals on the distorted fisheye image. The orientation of the object R_p relative to a virtual perspective camera frame centered on the apparent viewpoint can be calculated as a rotation correction to the object orientation R that is referenced to the global frame. The rotation correction matrix R_{adj} can be constructed as follows. First, the polar coordinates (θ_0, ϕ_0) of the intersection of the virtual camera optical axis with the sphere is calculated based on the 3D translation (x, y, z) of the object relative to the global frame

$$\theta_0 = \sin^{-1}\left(\frac{y}{\sqrt{x^2 + y^2 + z^2}}\right) \tag{11}$$

$$\phi_0 = \tan^{-1}\left(\frac{x}{z}\right). \tag{12}$$

The rotation adjustment matrix is then constructed column-wise using the coordinates of the rotated virtual camera frame axes in the global reference frame

$$X = [\cos\phi_0,\ 0,\ -\sin\phi_0] \tag{13}$$

$$Y = [-\sin\theta_0\sin\phi_0,\ \cos\theta_0,\ -\sin\theta_0\cos\phi_0] \tag{14}$$

$$Z = [\cos\theta_0\sin\phi_0,\ \sin\theta_0,\ \cos\theta_0\cos\phi_0] \tag{15}$$

$$R_{adj} = [X;\ Y;\ Z] \tag{16}$$

The orientation of the object relative to the virtual camera frame is then given as

$$R_p = R_{adj}R \tag{17}$$

The orientation branch of the network is trained to regress the apparent orientation R_p. The predicted true orientation R can be recovered using the predicted object translation by constructing the inverse R_{adj} matrix.

B. SilhoNet Adaptation to Fisheye

We compare three different variants of SilhoNet adapted for processing full fisheye images (Fig.1). For all variants, the size of the predicted silhouettes was increased to 128x128, because the handle objects in the UWHandles dataset have very thin features. The translation prediction output was also modified to predict the normalized pixel offset of the object center relative to the ROI directly without passing through a sigmoid function, and the predicted offsets were thresholded to lie within the ROI bounds. Because the dataset does not include segmentation annotations, the occluded

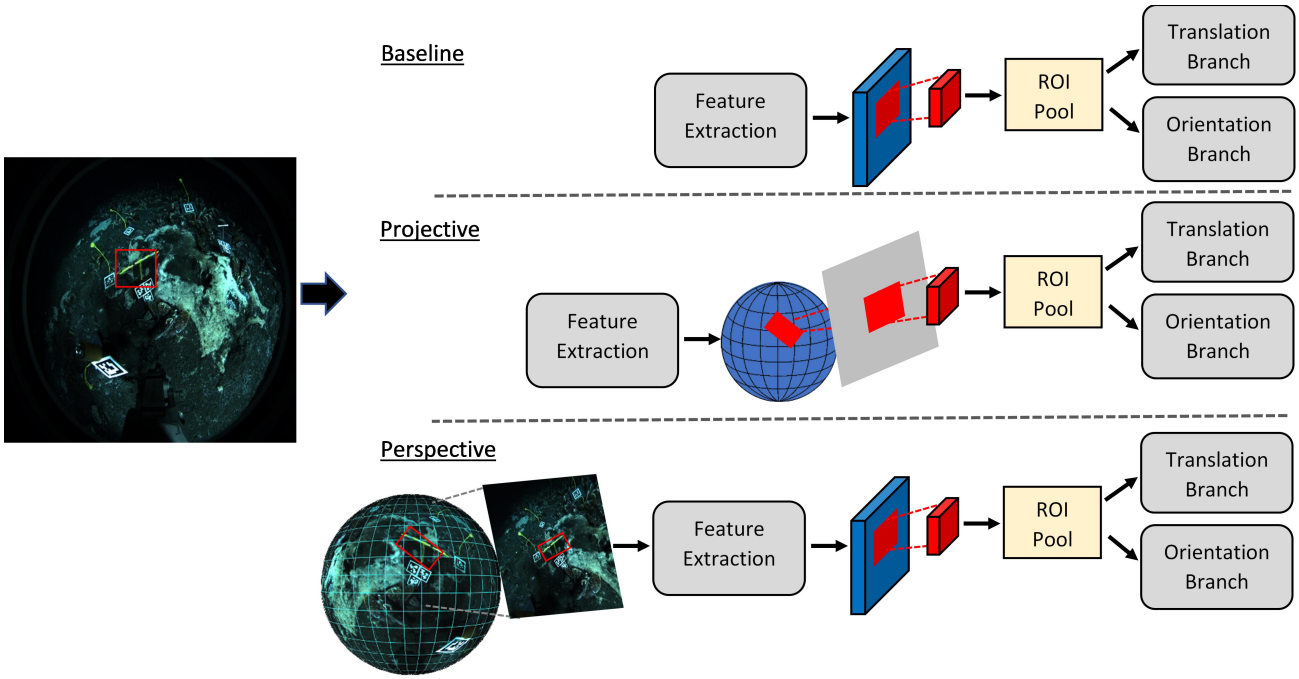

Fig. 1. Overview of the three different SilhoNet [1] adaptations for processing full fisheye images. The Baesline method processes the raw fisheye image directly through the unmodified network. The Projective variant processes the raw fisheye image through the feature extraction stage and then projects the features within the ROI through a spherical mapping to the tangent plane centered on the ROI, before processing the features through the ROI-pooling stage. The Perspective adaptation maps the fisheye image to a sphere and then generates a virtual perspective image for each object detection using a gnomonic projection, centered on the ROI. Each virtual image is then processed through the network.

Fig. 2. These objects have the same orientation relative to the rendered fisheye image frame but different translations, resulting in drastically different apparent orientations. Also, objects appear more distorted as they move from the image center.

silhouette branch was removed from the network. The orientation predictions of the handle objects were also reduced by their shape symmetries, as described in the SilhoNet paper [1]. Under these symmetry reductions, the network predicts orientations unique to shape symmetries only, which is appropriate for many object manipulation tasks, such as grasping tool handles, which are generally agnostic in feature space to how they are grasped. Also, because the symmetry reduction is applied directly to the training labels, no special care is needed to deal with symmetric objects in the training, and a simple distance loss function for orientation regression is used, as in the original method. The annotated ROIs were used as input to the network for both training and testing.

The first variant we consider as a baseline, which is essentially the vanilla SilhoNet architecture with the orientation branch output modified to regress the apparent orientation R_p, as described in the previous section. All variants of the network retain this prediction strategy. The second variant, which we refer to as "projective", processes the raw fisheye image through the feature extraction stage and then projects the features within the ROI through a spherical mapping to the tangent plane centered on the ROI, using the gnomonic projection. The projected features are then passed to the ROI-pooling stage. The motivating idea behind this projective strategy is that local features do not appear heavily distorted in fisheye images, but the spacial relationship of features across the ROI can be significantly distorted. The local feature map is thus generated directly on the raw fisheye image and then the spacial relationship of these local features is corrected through the projection onto the tangent plane. This projection operation is implemented as a Tensorflow layer for efficient and simple integration into the original network. The third variant, which we refer to as "perspective", projects a region of the fisheye image to a virtual perspective image centered on the ROI using the gnomonic projection. We chose the virtual image dimension to be 400x400 with a pixel relative focal length of 350. This virtual perspective image is processed through the feature extraction stage and then the ROI is cropped from the center of the feature map and passed to the ROI-pooling stage. Essentially, this method generates a virtual perspective image for each detected object and processes each of these virtual images separately through the network. This methods corrects for the fisheye distortions through the entire network pipeline. However, the computation scales with the number of detected objects, as a separate virtual image is processed for each one.

As a further comparison point, we take each of the three variants described above and replace the silhouette prediction branch with a branch that directly regresses the quaternion orientation, rather than first predicting a silhouette

and passing it to a second stage network for orientation prediction. This orientation branch has the same structure as the translation branch, but with the output size equal to 4x(# classes). The predicted quaternion for the class of the detected object is extracted from the output and normalized using an L2-norm. These methods which bypass the silhouette prediction to directly regress the orientation are referred to in the following sections by appending "_direct" to the name of the associated variant: "baseline_direct", "projective_direct", and "perspective_direct".

C. Network Training

The networks were trained with the same loss functions and dropout rates as in [1] on Titan V GPUs. All networks were trained for 400,000 iterations on the training set except for the "perspective_direct" method which was only trained for 356,000 iterations because of time constraints. Due to GPU memory limitations, the raw fisheye images of dimension 2,448x2,048 were downsampled by a factor of 3 for the baseline and projective variants and by a factor of 2 for the perspective variant. The baseline and projective variants were trained with a batch size of 2 and the perspective variant with a batch size of 3. As with the original SilhoNet method, the second stage network which regresses orientation from silhouettes was trained using only perfect rendered silhouettes.

IV. Dataset

We collected fisheye images of three different types of graspable handles, randomly arranged in different natural seafloor environments of the Costa Rican shelf break. Two of the handle types are actively used by Schmidt Ocean Institute and Woods Hole Oceanographic Institute to manipulate tools with Remotely Operated Vehicles (ROV) during underwater operations. AprilTag fiducials were randomly dispersed throughout the scene and on mount plates attached to the base of the handle objects in order to recover ground truth poses of the camera in the image sequences. The image sequences were then post processed with an annotation tool to obtain labeled 6D object poses and bounding boxes. The camera system was a FLIR BFLY-PGE-50S5C-C with a Fujinon FE185C086HA-1 fisheye lens centered in a dome housing that was mounted on the wrist of a Schilling Titan 4 hydraulic manipulator. This dataset is relevant to automating underwater manipulation tasks; if the pose of a handle attached to a known tool type can be accurately estimated, the handle can be autonomously grasped and manipulated to perform a desired manipulation task.

The dataset is composed of 25 training image sequences with a total of 18,329 images, 1 validation sequence with 910 images, and 2 testing sequences with 1,188 images.

A. Data Collection

We use AprilTag fiducials to obtain globally consistent camera poses in the image sequences. At various locations on the seafloor, the ROV was set down and the handle objects were randomly dispersed throughout the reachable area of the manipulator. 4 metal tag plates with attached 4" AprilTag fiducial stickers were randomly scattered around the handle objects. Initially, we also attached 4" April Tag stickers to mount plates at the base of the handle objects, but through trial and error, we found the most robust detection results were obtained with multiple 2" stickers attached to the object mount plates rather than the single 4" sticker. The smaller tags on the objects enabled better detection of the tags at close range, and the large tags on the scattered metal plates provided good detection at greater distances. While the pose of the handles could be directly recovered from the detected mounted tags if they remained rigid, we found that tag plates could move and sometimes break when handled with the hydraulic manipulator. Also, the mounted tags were not always visible or were poorly detected from certain camera viewpoints. Therefore, our annotation method assumes that all tag locations are static throughout an image sequence, but the exact transform between any tag and a tool handle is unknown and may change across sequences. Full 5MP resolution images were recorded at 3Hz, with the manipulator moving around the objects in various motion paths to obtain a diverse set of viewpoints.

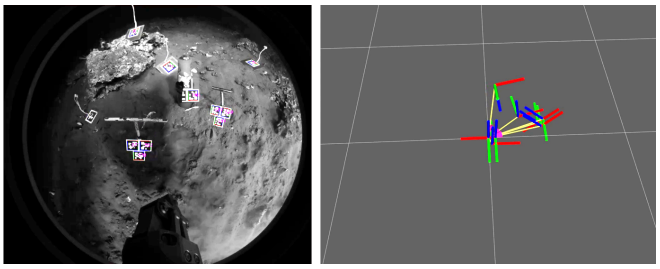

Fig. 3. Tags are detected in the raw fisheye images and processed through TagSLAM [40] to get globally consistent camera poses for an image sequence.

B. Data Processing

We used the ROS TagSLAM package [40] to process the image sequences and obtain globally consistent camera poses for each image in the sequence. In order to make use of the full fisheye view, tags were detected in the raw fisheye images (Fig. 3), and then the detected tag poses were calculated using a pinhole equidistant distortion model calibrated with the Kalibr ROS package [41], [42]. The pinhole model was adequate for this camera system, because the effective usable field of view of the image was less than 180°.

We created an OpenGL based annotation tool called VisPose, which takes in the image sequence with an associated camera pose file from the TagSLAM output. VisPose provides an interface to project models of the different objects into the image sequence, play through the sequence, and tweak the fit of the models to obtain accurate 6D pose annotations. Pose outliers can be filtered from the image sequence and then a COCO style annotation file exported, including 6D pose and 2D bounding box annotations, for the full image sequence (Fig. 4).

V. Results

The following section presents the performance of the different SilhoNet adaptations on the UWHandles dataset.

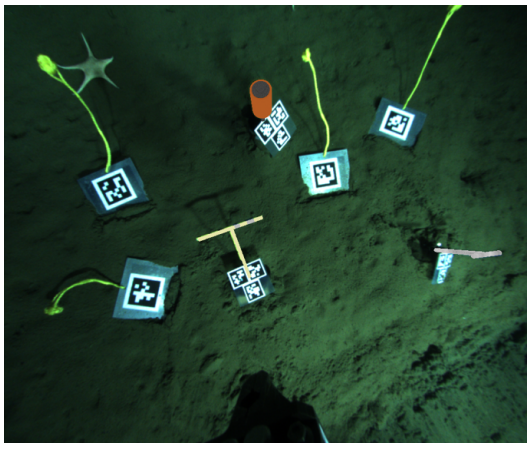

Fig. 4. Sample from the UWHandles dataset showing the handle object models projected into the fisheye image using the 6D pose annotations. The fisheye image is rectified here for visualization.

Table I and Table II show the percentage of translation and orientation predictions under different error thresholds, respectively. Table III shows the overall 6D pose prediction accuracy using the ADD-S metric from [43].

TABLE I

PERCENTAGE OF TRANSLATION PREDICTIONS UNDER THE THRESHOLD ERROR, WHERE A HIGHER PERCENTAGE UNDER A LOWER THRESHOLD MEANS BETTER ACCURACY.

Method	< 5cm	< 10cm	< 20cm	< 30cm
Baseline	69.88	90.81	98.48	99.92
Projective	71.91	87.35	96.22	98.39
Perspective	74.63	90.61	96.73	98.14
Baseline-Direct	47.55	72.04	94.56	99.21
Projective-Direct	46.71	73.56	93.54	98.36
Perspective-Direct	57.69	81.29	96.11	99.07

TABLE II

PERCENTAGE OF ORIENTATION PREDICTIONS UNDER THE THRESHOLD ERROR, WHERE A HIGHER PERCENTAGE UNDER A LOWER THRESHOLD MEANS BETTER ACCURACY.

Method	$< 5^\circ$	$< 10^\circ$	$< 20^\circ$	$< 30^\circ$
Baseline	12.75	34.77	62.78	75.31
Projective	11.26	35.33	63.03	74.89
Perspective	16.05	39.08	66.08	77.28
Baseline-Direct	22.58	45.58	69.00	81.31
Projective-Direct	28.91	50.45	69.48	83.19
Perspective-Direct	29.81	55.01	74.21	85.02

TABLE III

AREA UNDER ACCURACY-THRESHOLD CURVE FOR 6D POSE EVALUATION USING ADD-S METRIC FROM [43], WHERE A HIGHER AREA MEANS BETTER ACCURACY. PROJ. IS SHORT FOR PROJECTIVE AND PERSP. IS SHORT FOR PERSPECTIVE

Handle Type	Baseline	Proj.	Persp.	Baseline Direct	Proj. Direct	Persp. Direct
umichhandle	72.71	69.53	78.81	61.70	61.79	64.46
soihandle	71.48	79.98	75.11	47.51	53.33	60.77
whoihandle	61.82	57.34	61.95	48.54	47.39	59.90
ALL	68.65	68.92	71.94	52.57	54.15	61.69

For translation prediction errors under 5cm, which is a common measure of interest for pose estimation methods, the perspective variant shows significant performance improvement over the baseline method, while the projective method shows some improvement. All of the direct variants that remove the intermediate silhouette prediction branch show a drastic drop in translation prediction accuracy, indicating that even though the silhouettes are not directly used in the translation prediction, they enhance the networks ability to learn accurate feature scaling. The perspective-direct variant still shows significant improvement over the baseline-direct method, indicating that compensating for distortions in the fisheye image rather than directly predicting from the raw image is important for accurate pose predictions.

For orientation prediction errors under 5°, the perspective variant also shows significant performance improvement over the baseline method, while the projective variant does not perform as well as the baseline. In contrast to the translation predictions, all of the direct methods improve on the orientation prediction accuracy by approximately a factor of two across all variants, while the perspective-direct method still outperforms the baseline-direct method by a large margin. We observe that these initial results for orientation prediction fall short of the general target accuracy of less than 5deg error for manipulation applications. The UWHandles dataset is especially challenging for several reasons: the amount of training data is relatively small compared to terrestrial datasets, due to the expense of gathering underwater imagery; images are degraded by underwater back-scatter and lighting effects; the variance in camera viewpoints across an image sequence is high, due to the relatively low image collection frame-rate and large manipulator motions. Though these attributes make the dataset very challenging, they also motivate the development of methods that can work in real-world underwater environments with sparse training data. Future work will explore incorporating explicit methods of dealing with underwater effects, such as color correction and haze removal. We also note that the performance of the original SilhoNet [1] method was greatly improved through additional training on rendered data. Synthetically generated data can fill gaps in camera viewpoint representation missing in the real training data, allowing the network to better learn the full manifold of viewpoint representation. We are currently working on a method for synthetically rendering fisheye image data to supplement the UWHandles dataset. The synthetic images will be consistent with the real fisheye camera parameters, and the rendering process will incorporate some of the underwater imaging effects.

The ADD-S results also show a strong improvement in performance for the perspective variant against the baseline, both with and without the silhouette predictions, while the projective and baseline methods perform similarly. Because the ADD-S metric is generally most sensitive to translation errors, the results show stronger performance for the methods that retain the intermediate silhouette prediction over the direct methods. However, taking into account the separate orientation and translation results, better overall performance on this dataset might be achieved by a method which directly predicts the orientation but retains a silhouette prediction

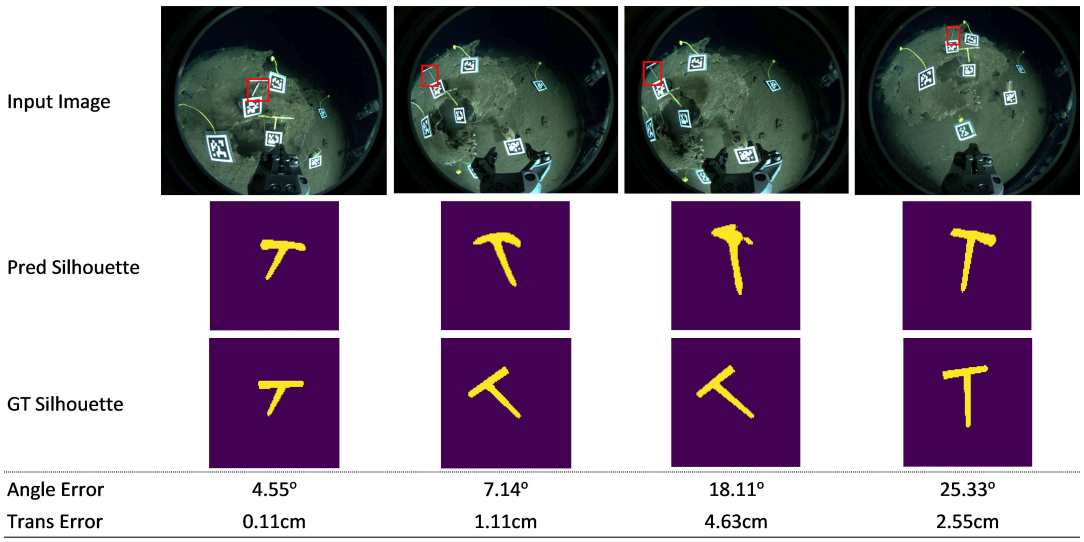

Fig. 5. Qualitative results with the perspective method on some sample test images for the whoihandle object. Predicted silhouettes and pose errors are shown for a range of errors from low to high.

branch during training to boost the translation accuracy. This is a method we plan to explore in the future. Overall, the results indicate that accounting for fisheye distortions before feature extraction, as the perspective method does, gives the best performance.

Figure 5 shows some qualitative results with the perspective method for the whoihandle object on some test samples, exhibiting a range of prediction errors. It is evident that the network successfully learns the silhouette representation of the handle object. However, some silhouette predictions are distorted or regress to offset viewpoints. We conjecture that these issues reflect the sparse coverage of the training data over the full viewpoint manifold of the objects and could be addressed through additional training on synthetic data, which will be part of our future work.

VI. Conclusion

In this paper, we presented a framework for adapting a ROI-based 6D object pose estimation method to work on full fisheye images. We demonstrated the adaptation of the SilhoNet [1] method on a new dataset of annotated fisheye images, called UWHandles, collected in natural underwater seafloor environments. The objects in the dataset are visually challenging handles, used in ROV operations to manipulate tools. The testing results on this dataset show that directly accounting for the fisheye distortions in the network before feature extraction is important for improving pose prediction accuracy, where the best performance was obtained with a method that generates a virtual perspective image centered on each ROI detection and processes these virtual undistorted images separately through the network. The results also show that the intermediate silhouette predictions of the SilhoNet method are important for the network to learn feature scaling to accurately predict translation. However, for this dataset, directly regressing the orientation rather than predicting from an intermediate silhouette achieves the greatest orientation accuracy. These observations motivate future investigation into a method which directly regresses the orientation, but retains the silhouette prediction branch during training to supervise the learning of feature scaling. Further improvements to the method will also be explored through explicit modeling of underwater effects in the image processing pipeline and augmentation of the training with synthetically rendered fisheye data.

References

[1] G. Billings and M. Johnson-Roberson, "Silhonet: An rgb method for 6d object pose estimation," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3727–3734, 2019.

[2] L. Deng, M. Yang, Y. Qian, C. Wang, and B. Wang, "Cnn based semantic segmentation for urban traffic scenes using fisheye camera," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 231–236.

[3] P. Goodarzi, M. Stellmacher, M. Paetzold, A. Hussein, and E. Matthes, "Optimization of a cnn-based object detector for fisheye cameras," in *2019 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, 2019, pp. 1–7.

[4] A. Sáez, L. M. Bergasa, E. Romeral, E. López, R. Barea, and R. Sanz, "Cnn-based fisheye image real-time semantic segmentation," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2018, pp. 1039–1044.

[5] J. Zhu, J. Zhu, X. Wan, C. Wu, and C. Xu, "Object detection and localization in 3d environment by fusing raw fisheye image and attitude data," *Journal of Visual Communication and Image Representation*, vol. 59, pp. 128–139, 2019.

[6] G. Salem, J. Krynitsky, M. Hayes, T. Pohida, and X. Burgos-Artizzu, "Cascaded regression for 3d pose estimation for mouse in fisheye lens distorted monocular images," in *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2016, pp. 1032–1036.

[7] Y. Shan and S. Li, "Discrete spherical image representation for cnn-based inclination estimation," *IEEE Access*, 2019.

[8] R. Monroy, S. Lutz, T. Chalasani, and A. Smolic, "Salnet360: Saliency maps for omni-directional images with cnn," *Signal Processing: Image Communication*, vol. 69, pp. 26–34, 2018.

[9] Q. Zhao, C. Zhu, F. Dai, Y. Ma, G. Jin, and Y. Zhang, "Distortion-aware cnns for spherical images.," in *IJCAI*, 2018, pp. 1198–1204.

[10] B. Coors, A. Paul Condurache, and A. Geiger, "Spherenet: Learning spherical representations for detection and classification in omnidirectional images," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 518–533.

[11] Y.-C. Su and K. Grauman, "Learning spherical convolution for fast features from 360 imagery," in *Advances in Neural Information Processing Systems*, 2017, pp. 529–539.

[12] ——, "Kernel transformer networks for compact spherical convolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9442–9451.

[13] C. M. Jiang, J. Huang, K. Kashinath, Prabhat, P. Marcus, and M. Nießner, "Spherical cnns on unstructured grids," in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, OpenReview.net, 2019.

[14] M. Eder and J.-M. Frahm, "Convolutions on spherical images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 1–5.

[15] Y. Lee, J. Jeong, J. Yun, W. Cho, and K.-J. Yoon, "Spherephd: Applying cnns on a spherical polyhedron representation of 360deg images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9181–9189.

[16] F. Zhao, S. Xia, Z. Wu, D. Duan, L. Wang, W. Lin, J. H. Gilmore, D. Shen, and G. Li, "Spherical u-net on cortical surfaces: Methods and applications," in *International Conference on Information Processing in Medical Imaging*, Springer, 2019, pp. 855–866.

[17] N. Perraudin, M. Defferrard, T. Kacprzak, and R. Sgier, "Deepsphere: Efficient spherical convolutional neural network with healpix sampling for cosmological applications," *Astronomy and Computing*, vol. 27, pp. 130–146, 2019.

[18] R. Khasanova and P. Frossard, "Graph-based classification of omnidirectional images," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 869–878.

[19] T. S. Cohen, M. Geiger, J. Köhler, and M. Welling, "Spherical cnns," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, OpenReview.net, 2018.

[20] P. Zhao, A. You, Y. Zhang, J. Liu, K. Bian, and Y. Tong, "Reprojection r-cnn: A fast and accurate object detector for 360°images," *arXiv preprint arXiv:1907.11830*, 2019.

[21] H. Feng, X. Yin, L. Xu, G. Lv, Q. Li, and L. Wang, "Underwater salient object detection jointly using improved spectral residual and fuzzy c-means," *Journal of Intelligent & Fuzzy Systems*, vol. 37, no. 1, pp. 329–339, 2019.

[22] Z. Chen, H. Gao, Z. Zhang, H. Zhou, X. Wang, and Y. Tian, "Underwater salient object detection by combining 2d and 3d visual features," *Neurocomputing*, vol. 391, pp. 249–259, 2019.

[23] D. Rathi, S. Jain, and S Indu, "Underwater fish species classification using convolutional neural network and deep learning," in *2017 Ninth International Conference on Advances in Pattern Recognition (ICAPR)*, IEEE, 2017, pp. 1–6.

[24] V. Lopez-Vazquez, J. M. Lopez-Guede, S. Marini, E. Fanelli, E. Johnsen, and J. Aguzzi, "Video image enhancement and machine learning pipeline for underwater animal detection and classification at cabled observatories," *Sensors*, vol. 20, no. 3, p. 726, 2020.

[25] D. A. Konovalov, A. Saleh, M. Bradley, M. Sankupellay, S. Marini, and M. Sheaves, "Underwater fish detection with weak multi-domain supervision," in *2019 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2019, pp. 1–8.

[26] W. Xu and S. Matzner, "Underwater fish detection using deep learning for water power applications," in *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*, IEEE, 2018, pp. 313–318.

[27] S. Marini, E. Fanelli, V. Sbragaglia, E. Azzurro, J. D. R. Fernandez, and J. Aguzzi, "Tracking fish abundance by underwater image recognition," *Scientific reports*, vol. 8, no. 1, pp. 1–12, 2018.

[28] A. Salman, S. A. Siddiqui, F. Shafait, A. Mian, M. R. Shortis, K. Khurshid, A. Ulges, and U. Schwanecke, "Automatic fish detection in underwater videos by a deep neural network-based hybrid motion learning system," *ICES Journal of Marine Science*, Feb. 2019.

[29] M. Moniruzzaman, S. M. S. Islam, M. Bennamoun, and P. Lavery, "Deep learning on underwater marine object detection: A survey," in *International Conference on Advanced Concepts for Intelligent Vision Systems*, Springer, 2017, pp. 150–160.

[30] J. Li, K. A. Skinner, R. M. Eustice, and M. Johnson-Roberson, "Watergan: Unsupervised generative network to enable real-time color correction of monocular underwater images," *IEEE Robotics and Automation letters*, vol. 3, no. 1, pp. 387–394, 2017.

[31] C. Li, J. Guo, and C. Guo, "Emerging from water: Underwater image color correction based on weakly supervised color transfer," *IEEE Signal processing letters*, vol. 25, no. 3, pp. 323–327, 2018.

[32] H. Kuang, Q. Xu, and S. Schwertfeger, "Depth estimation on underwater omni-directional images using a deep neural network," in *The International Conference on Robotics and Automation*, *Workshop on Underwater Robotics Perception*, 2019.

[33] M. Jeon, Y. Lee, Y.-S. Shin, H. Jang, and A. Kim, "Underwater object detection and pose estimation using deep learning," *IFAC-PapersOnLine*, vol. 52, no. 21, pp. 78–81, 2019.

[34] M. C. Nielsen, M. H. Leonhardsen, and I. Schjølberg, "Evaluation of posenet for 6-dof underwater pose estimation," in *OCEANS 2019 MTS/IEEE SEATTLE*, IEEE, 2019, pp. 1–6.

[35] J. Fu, I. V. Bajić, and R. G. Vaughan, "Datasets for face and object detection in fisheye images," *Data in brief*, vol. 27, p. 104 752, 2019.

[36] S. Yogamani, C. Hughes, J. Horgan, G. Sistu, P. Varley, D. O'Dea, M. Uricár, S. Milz, M. Simon, K. Amende, *et al.*, "Woodscape: A multi-task, multi-camera fisheye dataset for autonomous driving," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9308–9318.

[37] (2019). Fisheye projection, [Online]. Available: `https://wiki.panotools.org/Fisheye_Projection`.

[38] M. Cowlishaw. (2014), [Online]. Available: `http://speleotrove.com/pangazer/gnomonic_projection.html`.

[39] E. W. Weisstein. (2020). Gnomonic projection, [Online]. Available: `http://mathworld.wolfram.com/GnomonicProjection.html`.

[40] B. Pfrommer and K. Daniilidis, "Tagslam: Robust slam with fiducial markers," *arXiv preprint arXiv:1910.00679*, 2019.

[41] J. Maye, P. Furgale, and R. Siegwart, "Self-supervised calibration for robotic systems," in *2013 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2013, pp. 473–480.

[42] P. Furgale, J. Maye, and J. Rehder. (2014). Ethz-asl/kalibr, [Online]. Available: `https://github.com/ethz-asl/kalibr/wiki`.

[43] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," in *Proceedings of Robotics: Science and Systems*, Pittsburgh, Pennsylvania, 2018.