

# Machine Learning for Autonomous Jellyfish Orientation Determination and Tracking

Adam Wiktor  
SUNet ID: 005550797

**Abstract**—Automated jellyfish classification and orientation determination can be used to improve the robustness of an autonomous jelly tracking system. A set of training images were segmented and augmented with orientation data, and a feature vector was generated for each image. An SVM was then trained to estimate a jelly’s orientation. In simulations on unseen test images, the approach was able to correctly determine the orientation of a jellyfish to within  $\pm 15^\circ$  in 68% of cases. This is improved to within  $\pm 15^\circ$  in 96% of cases on a training set. The algorithm was successful at categorizing the 3D angle of the jelly out of the image plane, and in certain cases was also able to distinguish between a jelly and a foreign object in the image frame.

## I. INTRODUCTION

Research performed at the Monterey Bay Aquarium Research Institute (MBARI) on gelatinous marine animals (commonly known as Jellyfish, abbreviated Jellies here) frequently requires that observations be taken over a long period of time. This is typically done using a Remotely Operated Vehicle (ROV) driven by a human pilot, but the concentration required to continually deliver precise thruster control can be tiring for the pilot[3]. A semi-autonomous pilot aid for jelly tracking has been demonstrated to greatly reduce fatigue, allowing observation runs to last as long as 90 minutes[2]. Farther in the future, a more robust tracking system could eventually allow a fully automated Autonomous Underwater Vehicle (AUV) to follow jellies over many hours or days, collecting valuable data in all different environmental conditions.

The pilot aid system previously tested at MBARI relies on stereo vision and blob-tracking techniques. To differentiate between jellies and the surrounding ocean, a grayscale image is first thresholded to detect all light-colored blobs[1]. After filtering the blobs by pixel area to eliminate marine snow, each jelly is tracked by the coordinates of the blob centroid. Basic target recognition is performed by monitoring the apparent size of the blob and its distance from the camera, enabling the system to continue tracking the target in a few cases where other objects entered the frame[1]. A control law uses this information along with a dynamic model of the vehicle to attempt to maintain the relative position between the ROV and the jelly. However, no attempt has been made to date to account for the dynamics and motion of the jelly itself.

Past experiments demonstrated that many of the tracking failures occurred either due to the target moving out of the camera frame, or else from a loss of target recognition when other animals pass near the jelly[2], [1]. Machine learning techniques offer the possibility to improve the robustness of the system in both areas. In the first case, machine learning algorithms can estimate the orientation of the jelly in the



Fig. 1: Sample *paraphyllina* image (MBARI).

image, allowing the control law to account for its potential future motion. Jellyfish are able to accelerate, swim, and change orientations, but in general at any given time will move only in an axial direction. Therefore, if a machine learning algorithm is able to supply an orientation estimate, the ROV will be able to follow the path of a jelly even if it leaves the camera frame. The angle of the jelly from the vertical direction is critical to know the direction that the cameras should pan if the target is lost. However, a 3D orientation estimate is also desired because the cameras can lose track of the jelly if it gets too close. Therefore, an angle into or out of the image plane would help predict if the ROV will need to advance or back up. Similarly, machine learning also offers the possibility of improving the target tracking algorithms to reject other objects that enter the camera frame. By matching the features of the target jelly against a library of similar animals (or the same jelly at earlier times), the system would be able to continue following the target even after interferences that ended previous experiments[1].

This project presents several new machine learning techniques to improve the robustness of the jelly tracking system. First, machine learning can be used to improve the segmentation processes used to isolate the target in an image frame. An algorithm to autonomously estimate the 3D orientation of a jelly is also outlined. To demonstrate the technique, a single genus - *paraphyllina* - was selected due to its clear axial features (Figure 1). MBARI supplied a set of several hundred images from past ROV dives showing *paraphyllina* in various orientations. This database is split into a set of training images and 100 test images, and cross validation is used to demonstrate the applicability of the method to new, unseen jelly images. The algorithm uses several techniques to generate a feature vector for each image, which can then be used in a

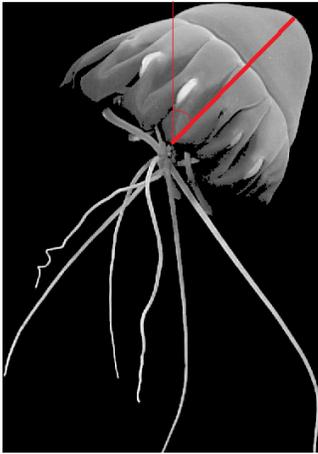


Fig. 2: Jelly orientation. A manual estimate of the jelly’s orientation is shown in red. The estimated orientation for this example is  $44^\circ$  (category 2) with a depth of  $0^\circ$ .

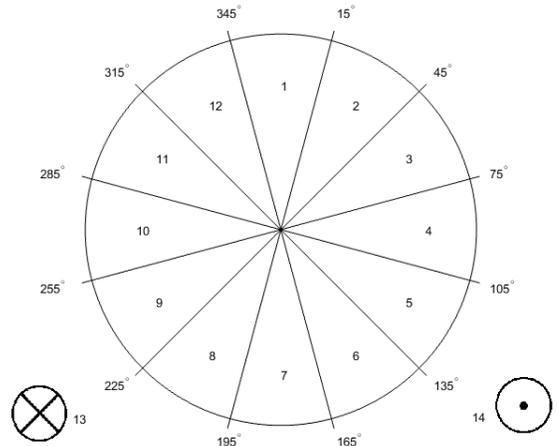


Fig. 3: The orientation is bucketed into one of 14 categories, 1-12 for the in-plane angle from vertical, and 13-14 for the 3D out-of-plane angle.

standard multi-class classification scheme to categorize the 3D orientation of the jelly. Finally, the same feature vectors can be used to distinguish between jelly and non-jelly objects in a frame to isolate the target and reliably track it in a wide range of environments.

## II. APPROACH

### A. Image Processing

MBARI provided a set of several thousand images of jellies taken from previous research dives. After sorting the images and removing any with a target too small or out of focus, 140 usable images of *paraphyllina* in various orientations remained. This training set was augmented by rotating each image by  $90^\circ$ ,  $180^\circ$ , and  $270^\circ$  to provide additional orientations on which to train the algorithm, for a total database of  $m = 560$  images. Orientation data was manually added to each image by drawing an estimate of the jelly’s axis on the frame, and an estimated depth angle was entered as either  $-90^\circ$  (jelly pointed directly away from camera),  $-45^\circ$ ,  $0^\circ$ ,  $45^\circ$ , or  $90^\circ$  (jelly pointed directly towards the camera) (Figure 2). The angle data for each image was then bucketed into 14 categories  $\beta$  ( $30^\circ$  each, plus out-of-frame and into-frame, see Figure 3), and stored as a matrix showing the true orientation for the image:

$$Y = \begin{bmatrix} -y^{(1)T} \\ \vdots \\ -y^{(i)T} \\ \vdots \\ -y^{(m)T} \end{bmatrix}; y^{(i)} = \begin{bmatrix} 1\{\beta_i = 1\} \\ \vdots \\ 1\{\beta_i = k\} \\ \vdots \\ 1\{\beta_i = 14\} \end{bmatrix}$$

Two separate image segmentation methods were tested for comparison. First, a threshold-based blob detection algorithm similar to [1] was implemented in MATLAB. The image was converted to grayscale, and all pixels below an individually-tuned threshold were considered background and discarded. The thresholded image was then passed to a blob detection

routine that finds the centroid and area of each light region in the image. All blobs with an area below a preset minimum are rejected to eliminate marine snow, and the largest remaining blob is stored as the jelly.

This technique works well for a large number of images, such as Figure 2. However, by converting the image to grayscale at the start, this method effectively discards two-thirds of the usable information in the frame. In certain situations, the brightness of the jelly pixels can be very close to the intensity of the surrounding ocean, and color is actually a much better indicator. For this reason, an alternative segmentation technique based on machine-learning was also tested. In this method, an Expectation-Maximization (EM) algorithm is used on the full-color image to model the pixel colors as being drawn from two separate probability distributions (one corresponding to the jelly, and the other for the ocean). Every pixel color is treated as a point in 3D space, and the algorithm finds a mean and covariance for the two distributions. Finally, each pixel is assigned a label based on the likelihood of it being drawn from either distribution. Figure 4 shows the red and blue pixel values, along with the two probability distributions found by the EM algorithm. There is still an ambiguity in determining which of the two distributions corresponds to the jelly. This ambiguity can be overcome using prior knowledge, such as the fact that the ocean pixels have more blue content, or the fact that the jelly almost always occupies fewer than half of the pixels in a frame. Finally, once the pixels have been labeled and the background pixels discarded, the same blob tracking method can be used to remove marine snow and isolate the jelly. Figure 5 shows a comparison of the thresholded and EM-based segmentation methods.

### B. Orientation Classification

To train a classifier to recognize the orientation of a jelly, each image  $i$  must be represented as a feature vector  $\phi_i$  that describes the elements of the image. This feature vector can be generated in several ways. In the first method, the images are first sorted by their orientation category. Several thousand features are then generated for each image using the

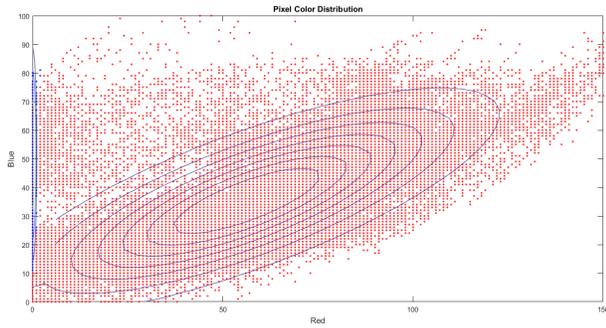


Fig. 4: Red and blue content of all pixels in a sample image. Blue points were categorized by the EM algorithm as “background”, and red points were categorized as “jelly”. Contour lines for the two probability distributions are also shown.

Speeded Up Robust Features (SURF) algorithm. However, not all features generated from SURF will be useful for orientation prediction. Therefore, the K-means algorithm is used to reduce the number of features across all categories and find the features most highly associated with the jellies’ orientation. The K-means approach reduces the feature set to a 500-element vector for each image.

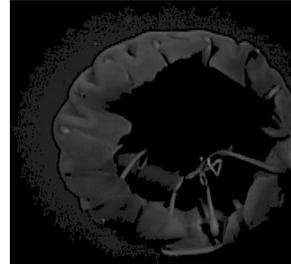
Once the feature vectors  $\phi_i$  have been generated by the K-means algorithm, an SVM is trained on the data. Cross validation is used to ensure that the model is not overfitting the training data. The set of 560 images is divided with 80% in a training set and 20% in the test set. Using a one-vs-all approach, a hypothesis function  $h_{\theta_k}(x^{(i)}) = \theta_k^T \phi(x^{(i)})$  is trained on this training set. The SVM produces a vector  $\theta_k$  for each of the 14 orientation categories. To make a prediction for an image  $x^{(i)}$ , the margin for each category  $k$  is calculated. The algorithm ultimately selects the category with the highest probability as the most likely orientation:  $k_{prediction} = \underset{k}{\operatorname{argmax}} \theta_k^T \phi(x^{(i)})$ .

An alternative approach to image classification relies on neural networks. For comparison with the procedure outlined above, a neural network was trained using MATLAB’s neural network toolbox on the thresholded images (with no orientation data). This neural network generated a 50-element feature vector  $\phi_i$  describing each image. A softmax function then uses this set of features  $\Phi$  along with the orientation data  $Y$  to make a classification prediction. To improve the features generated for each image, the neural network can then be fine-tuned with a second pass using the prediction errors from the first softmax function. Similar to the approach above, the most likely orientation category is returned.

Either method outlined above can also be extended to other aspects of the jelly tracking system. For example, when a second object enters the image frame, the algorithm can estimate which of the two is a jelly by attempting to predict an orientation for both. The algorithm will have higher confidence for the object with features most similar to the training images. Therefore, this single classifier can serve as a rudimentary approach to differentiating between multiple objects in a frame simply by comparing the orientation probabilities for each.



(a)



(b)



(c)

Fig. 5: Image segmentation methods. (a) Raw image. (b) Jelly segmented using a grayscale threshold. (c) Jelly segmented using EM-based technique.

### III. RESULTS

#### A. Image Processing

The EM-based image segmentation technique was highly successful at differentiating between the jelly and the surrounding ocean. Figure 5 shows an example where thresholding removed a significant portion of the jelly features, and still failed to completely remove the background. The EM approach, on the other hand, removed all of the background without significantly affecting the jelly. The downside is that the EM algorithm required significantly more time to run since it had to perform up to 20 iterations on thousands of pixels. However, this would not necessarily be a significant problem in practice since the color distribution does not change significantly from one frame to the next. The EM algorithm could therefore be run only once every few seconds to generate new estimates for the mean and covariance, and simply use the previous values for each frame in between.

Other image processing techniques also helped increase the classification accuracy. As Figure 6 shows, the strongest features generated by the SURF algorithm are on the tentacles of the jelly and not on its body. However, these tend to be poor predictors of orientation since the tentacles can point in arbitrary directions. Removing the tentacles from the image using MATLAB’s *imerode()* and *imdilate()* commands on the segmentation mask increased the accuracy of the classifier by roughly 10%.

#### B. Orientation Classification

The K-means approach was successful at categorizing the jelly orientation for most images. When applied to the training data, the algorithm categorized the orientation correctly for

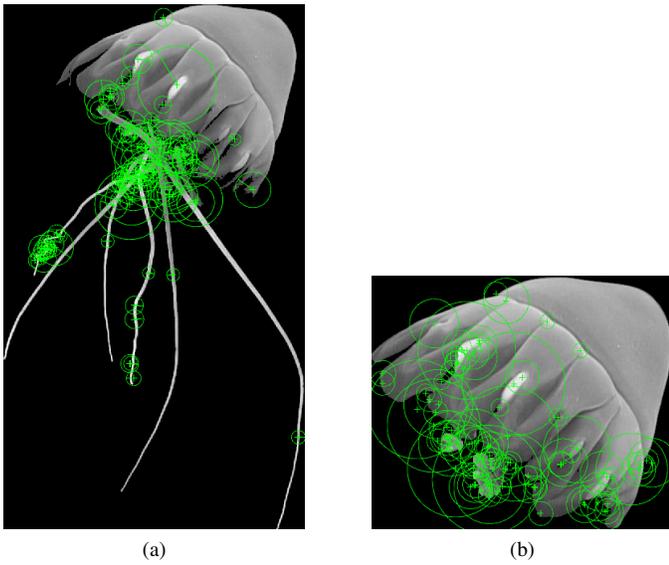


Fig. 6: 50 strongest features found by the SURF algorithm. (a) Jelly image with tentacles. (b) Jelly image with tentacles removed.

96% of the images. The classifier also performed well on the test set, correctly classifying 68% of the images. More importantly still, even in situations when the classifier made an error, it tended not to predict orientations too far from the correct class. In many cases, determining the exact orientation of the jelly is less important than knowing the quadrant in which it is likely to be found if it leaves the frame. Therefore, a jelly that the classifier labels as being in an adjacent category may also be considered successful. For example, if the true jelly orientation is in the  $-15^\circ$  to  $15^\circ$  category, but the classifier labels it as  $16^\circ$  to  $45^\circ$ , the camera still knows the approximate direction in which it should pan, so the label may be considered a partial success. By this measure, the classifier was successful on 99.7% of the training images and 88% of the test images.

The 3D category prediction was also relatively successful, although it had significantly fewer images on which to train since it wasn't possible to increase the data set by rotating these images. The out-of-plane angle is difficult to measure even for a human observer, so the 3D classifier focused simply on categorizing jellies pointed almost directly into or out of the plane (such as Figure 5) rather than trying to estimate an exact angle. This is still useful information for the control system, though, since the ROV may need to move to the side in situations where the jelly is pointed directly towards it to avoid making contact and disturbing the jelly. Using this measure, the classifier was successful at categorizing the 3D angle for 94% of images. Note that in this case the data set was too small to split into a complete training and test set, so the accuracy is only reported for the training set.

Extending the K-means/SVM classifier to other problems was somewhat successful. In test cases where the orientation SVM was used to differentiate between a jelly and a non-jelly object (for example, a piece of seaweed), the prediction confidence was significantly higher for the jelly, so the method

Output Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1	64 16.0%	1 0.3%	0	0	0	0	0	0	0	0	0	1 0.3%	1 0.3%	0	95.5% 4.5%
2	0	12 3.0%	0	0	0	0	0	0	0	0	0	0	0	0	100% 0.0%
3	0	0	16 4.0%	0	0	0	0	0	0	0	0	0	0	0	100% 0.0%
4	0	0	4	62 15.3%	3 0.8%	0	0	0	0	0	0	0	0	0	89.0% 10.1%
5	0	0	0	0	7 1.8%	0	0	0	0	0	0	0	0	0	100% 0.0%
6	0	0	0	0	0	9 2.3%	0	0	0	0	0	0	0	0	100% 0.0%
7	0	0	0	0	0	1	64 16.0%	3 0.8%	0	0	0	0	0	0	94.1% 5.9%
8	0	0	0	0	0	0	0	17 4.3%	0	0	0	0	0	0	100% 0.0%
9	0	0	0	0	0	0	0	0	11 2.8%	0	0	0	0	0	100% 0.0%
10	0	0	0	0	0	0	0	0	0	66 16.5%	3 0.8%	0	0	0	95.7% 4.3%
11	0	0	0	0	0	0	0	0	0	0	19 4.8%	0	0	0	100% 0.0%
12	0	0	0	0	0	0	0	0	0	0	0	21 5.3%	0	0	100% 0.0%
13	0	0	0	0	0	0	0	0	0	0	0	0	7 1.8%	0	100% 0.0%
14	0	0	0	0	0	0	0	0	0	0	0	0	0	8 2.0%	100% 0.0%
	100%	92.2% 7.7%	80.0% 20.0%	100%	70.0% 30.0%	93.0% 7.0%	100%	85.0% 15.0%	100%	100%	98.4% 1.6%	95.5% 4.5%	87.5% 12.5%	100%	95.8% 4.2%
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	

Fig. 7: Training images confusion matrix.

Output Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1	15 15.0%	1 1.0%	0	0	1 1.0%	0	0	0	0	0	0	1 1.0%	1 1.0%	0	78.9% 21.1%
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	NaN% NaN%
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	NaN% NaN%
4	0	1 1.0%	3 3.0%	16 16.0%	1 1.0%	0	0	0	0	0	0	0	1 1.0%	0	72.7% 27.3%
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	NaN% NaN%
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	NaN% NaN%
7	0	0	1 1.0%	0	0	2 2.0%	16 16.0%	5 5.0%	1 1.0%	0	0	0	0	0	64.0% 36.0%
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	NaN% NaN%
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	NaN% NaN%
10	0	1 1.0%	1 1.0%	0	1 1.0%	0	0	1 1.0%	16 16.0%	4 4.0%	1 1.0%	0	1 1.0%	0	61.9% 38.1%
11	0	0	0	0	0	0	0	1 1.0%	0	2 2.0%	1 1.0%	0	0	0	50.0% 50.0%
12	1 1.0%	0	0	0	0	0	0	0	0	0	2 2.0%	0	0	0	66.7% 33.3%
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	NaN% NaN%
14	0	0	0	0	0	0	0	0	0	0	0	0	0	1 1.0%	100% 0.0%
	83.3% 16.7%	100%	100%	100%	100%	100%	100%	100%	100%	33.3% 66.7%	40.0% 60.0%	100%	50.0% 50.0%	100%	68.0% 32.0%
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	

Fig. 8: Test images confusion matrix.

would be able to reject the other object. However, when comparing an image of *paraphyllina* with some other type of jelly, the classifier's confidence in the orientation prediction was a poor indicator of which jelly to follow. Many other factors, such as lighting, view angle, and distance, tended to have more of an effect on margin than the type of jelly. Therefore, while simply comparing the confidence in an orientation prediction might have some value for rejecting foreign objects, a second classifier trained to distinguish between the target jelly and alternate marine animals would be more effective

The algorithm was also able to classify the orientation

using the alternative neural network approach. In general, though, this did not perform as well as the previous method. A neural network trained with a softmax function and fine-tuned with back propagation determined the correct labeling for 80% of the training set, but only 40% of the test set. Using the within-one-category criteria, the neural network was successful at labelling 89% of the training set and 71% of the test set. Besides being less accurate than the K-means approach, the neural network also has the disadvantage that it does not generate a useful feature vector and instead operates as a “black box” that simply inputs an image and returns a predicted class. Attempts to train a classifier using a feature vector output by the neural network were not successful. An SVM trained on these feature vectors only achieved 20% accuracy. This is better than random chance (8%), but it is not accurate enough for practical classification. This limits the usefulness of the approach since it can no longer be extended to other aspects of the jelly tracking problem.

#### IV. FUTURE WORK

The main task remaining is to fully implement the algorithm to classify jelly orientations from video frames in real time. This is necessary before the approach can be tested on an ROV for use in a control architecture. The image encoder can be stored, so it does not need to re-run the full k-means analysis for each new image. The trained SVM can also be stored, which should speed up the process enough to take place in near real time. While further improvements to the test image classifier accuracy are desirable, the current architecture offers enough accuracy to deal with the relatively rare failure scenarios in the existing control system.

There are several avenues that could also be explored to improve the current classification scheme. For example, the classifier could be extended to take previous orientations into account. Since jellies usually do not move very quickly, cases with low prediction confidence could be improved by considering a previous estimate with higher margin and using that data to guide the prediction. Another approach would be to attempt to directly model the orientation of a jelly from the feature set without reducing it into discrete buckets. This has the advantage of allowing for a loss function that heavily penalizes orientation estimates farther from the true orientation. Following this approach could potentially improve the number of images correctly classified in the “within-one-category” region.

Finally, a second classifier could be implemented to help distinguish between a target jelly and other objects that may enter the frame. This will require modifying the blob tracking algorithm to follow multiple objects, but otherwise the approach is identical to the orientation classification technique demonstrated here. Once a specific type of jelly has been targeted, a classifier trained to recognize that genus could be used to reject all other animals that enter the frame. Using this approach, a single routine will ultimately be able to take in a new image, classify it as target/other, estimate the 3D orientation of the jelly, and then pass all this information to the controller to enable a highly robust tracking system.

#### V. CONCLUSION

Using a training set of images, two methods for segmenting the image to isolate the jelly have been implemented. A grayscale threshold-based approach was functional for a majority of the images, but an approach that uses the EM algorithm to estimate the mean and covariance of the color distributions was demonstrated to be more effective. This method had a slower run time, but had the advantage of being able to distinguish between the target and the background for darker images where thresholding failed.

Machine learning techniques were also effective for estimating the orientation of a jelly from an image. By training an SVM on features generated using SURF and K-means algorithms, the approach was able to correctly classify the orientation for 96% of training images and 68% of test images. The 3D angle out of the plane was also included in the data, and the classifier successfully distinguished between jellies pointed towards, away, and normal to the camera. Finally, the classifier was able to roughly distinguish between jellies and other objects using the confidence in its orientation predictions. A neural network approach was also implemented, but it was significantly less accurate than this method and therefore will not be pursued further.

Once implemented on an ROV, these improvements will make a significant difference to the robustness of the jelly tracking system. The new image segmentation technique could allow an existing control architecture to get a more accurate estimate of the position and shape of the jelly, allowing for better tracking as outlined in [2]. An orientation estimate generated using this algorithm will then allow for a more advanced control law that can take the jelly’s current and projected motion into account. Finally, a method to distinguish between the target and other objects would also fix the failure modes that ended several of the previous tracking trials. All together, these improvements could dramatically increase the observation time of the jelly tracking system and provide unprecedented data for marine scientists.

#### VI. ACKNOWLEDGMENTS

This work would not have been possible without the assistance and training images from the Monterey Bay Aquarium Research Institute. Prof. Stephen Rock (Stanford University Department of Aeronautics and Astronautics) and the members of the Aerospace Robotics Lab also contributed invaluable advice and feedback.

#### REFERENCES

- [1] Rife, J.H., Rock, S.M.: Segmentation methods for visual tracking of deep-ocean jellyfish using a conventional camera. *IEEE JOURNAL OF OCEANIC ENGINEERING* **28.4** (2003)
- [2] Rife, J.H., Rock, S.M.: Design and validation of a robotic control law for observation of deep-ocean jellyfish. *IEEE Transactions on Robotics* **22.2** (2006)
- [3] Rife, J.H., Rocke, S.M.: A pilot-aid for roV based tracking of gelatinous animals in the midwater. In: *OCEANS, 2001. MTS/IEEE Conference and Exhibition*, vol. 2, pp. 1137 – 1144 vol.2 (2001). DOI 10.1109/OCEANS.2001.968274