B.Comp. Dissertation

Underwater Real-Time Object Recognition and Tracking for Autonomous Underwater Vehicle

By

Tan Soon Jin

Department of Computer Science

School of Computing

National University of Singapore

2016/17

B.Comp. Dissertation

Underwater Real-Time Object Recognition and Tracking for Autonomous Underwater Vehicle

By

Tan Soon Jin

Department of Computer Science

School of Computing

National University of Singapore

2016/17

Project No: H021400
Advisor: Prof. Terrence Sim Mong Cheng
Deliverables:
    Report: 1 Volume

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background on Robosub

1.1.1 Information about the competition

Robosub is an international AUV competition where students from around the world build their own customized AUV to complete a series of underwater missions that involve both visual tasks and acoustics task. The competition is held annually in TRANSDEC (Transducer Evaluation Center) man-made pool.

Figure 1.1: Aerial view of TRANSDEC. Operational depth of 16 ft for most vision tasks

1.1.2 Description of vision tasks

Vision tasks in Robosub can divided into forward-facing tasks and bottom-facing tasks which poses different sets of challenges. Since the tasks do not vary significantly every year, we can use datasets collected from this year's competition as testbed for our vision algorithms.

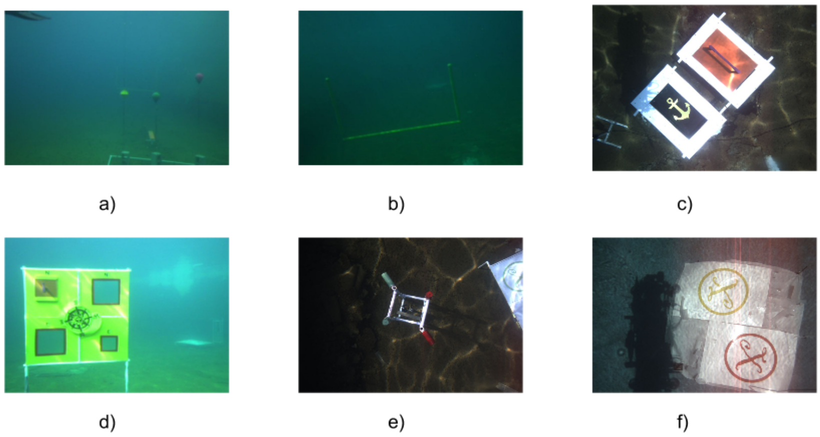


Figure 1.2: Robosub 2016 Vision Tasks. a) Scuttle Ship b) Navigate Channel c) Weigh Anchor d) Set Course e) Bury Treasure (Coins) f) Bury Treasure (Island)

1. **Scuttle Ship (Buoy)** A recurring task where the AUV has identify the correct color buoy and touch it. There are two major challenges with this task:

   a. Red buoy tends to exhibit color distortion as red wavelength attenuates the fastest (Galdran, Pardo, Picón, & Alvarez-Gila, 2015).

   b. Non-uniform illumination on top-half of buoys make it hard to distinguish the buoys.

2. **Navigate Channel**

   The AUV is required to move in between and over the PVC pipes.

3. **Weight Anchor**

   Classic object classification task where the AUV is required to drop a marker into the correct bin to obtain maximum points after removing the cover using a manipulator.

4. **Set Course**

   Identification of covered square (orange panel) and remove it. Fire two markers over 2 smaller holes. As yellow and orange are really close on the colour spectrum, this forces us to use other visual cues such as edge for better detection.

5. **Bury Treasure**

   For this task, one has to identify the small cylinders (red and green) and drop them onto their respective colored circles (on the Island). Identifying and distinguishing small objects afar (4 m) underwater is the biggest challenge in this task. Besides that, the dropped cylinders may potentially occlude the circles.

1.2 Challenges in Underwater Image Processing

Many literature such as M, Abhilash, and Supriya (2016) that investigates various underwater image restoration methods cite haze formation which happens as light propagated from object undergoes attenuation and scattering causing image with low contrast. In addition, Beer-Lambert law (Gevers, Gijsenij, Van de Weijer, & Geusebroek, 2012) states relates attenuation of light to properties of water medium; therefore, light components with low wavelength; green and blue are not as easily absorbed compared to red wavelength. This causes underwater images tend to have greenish or bluish color cast.

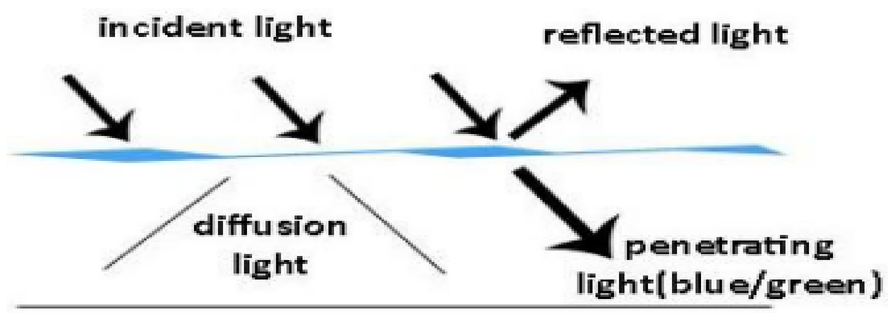


Figure 1.3: Absorption of light at the surface

1.3 Project Requirements Analysis

Though it is the objective of the project to design a vision framework for the Robosub missions, the vision framework should also be easily extended to work for more complex real world applications.

1.3.1 Nature of tasks

1. Vision algorithms perform with acceptable accuracy under the following conditions:

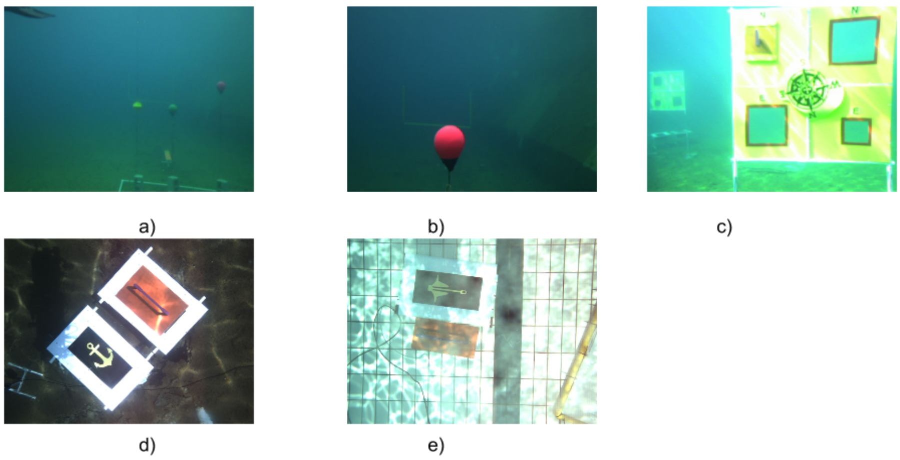


Figure 1.4: Different vision challenges. a) Haze formation b) Partial occlusion c) Non-uniform illumination d) Sunlight flickers e) Shadow

2. Low detection latency (near real-time)

   AUV needs to make swift decision based on sensor inputs to complete task under time constraints (same for real world time critical mission i.e underwater mine detection)

3. Geometric properties of objects are made known in advance

4. Short-period single target tracking for task (unlike video surveillance application)

5. Able to detect objects from far away (5m) and near distance (for manipulation task)

Chapter 2

Literature Review

This review is conducted with the purpose to investigate and select most suitable algorithms that generate the best result on the Robosub datasets. Since every teams who participate in Robosub are required to submit a journal paper,vision algorithms deployed by top-peforming schools such as Cornell University, University of Florida and cole de technologie suprieure provide valuable insights on image processing that are effective in underwater environment. Besides that, review of popular image processing techniques in particular on topics like object detection, object tracking, color constancy, saliency mechanism, detection proposals and adapatation of algorithms.

2.1 Preprocessing

2.1.1 Underwater Image Enhancement

The paper by Garcia, Nicosevici, and Cufí (2002) compared methods such as homomorphic filtering and local adaptive histogram equalization (Contrast Limited Adaptive Histogram) which considers that image is a product of illumination and reflectance properties. However, homomorphic filter has the benefit of preserving sharp edges while attenuating non-uniform illumination. On the other hand, by only redistributing pixels exceeding a clipping level to increase contrast of an image, CLAHE manages to reduce noise amplification in normal local histogram equalization.

Instead of relying on a single image, Gracias, Negahdaripour, Neumann, Prados, and Garcia (2008) recover corrupted underwater image by finding the difference between the current frame with temporal median of a registered set of N frames. Image dehazing is equally as important to ensure good performance of further image processing operation such feature detection. Kaiming, Jian, and Xiaoou (2011) proposed a single image dehazing method using the dark channel prior which states that haze-free image contains local region with low intensities in at least one color channel. Galdran et al. (2015) propose a variant of dark channel prior for underwater environment, the Red Channel method as red color shows most degradation in turbid water medium. From another perspective, Ancuti and Bekaert (2011) takes a fusion-approach to recover the original image by generating a few weight maps that correlates with intrinsic properties of the image itself. A color corrected and contrast enhanced of the input image are used to generate different weight maps that are fused using a Laplacian multi-scale strategy to generate a smoothed output image. This method has the benefit of using a single image but the weight maps must be combined with different weightage to achieve an ideal result.

2.1.2 Color Constancy

Color cue plays an important role to distinguish different objects such as the small cylinders in Robosub that requires sorting by color. The ability to account for color of the light source is called color constancy. The work of Gijsenij, Gevers, and Van De Weijer (2011a) analyzes various color constancy algorithms. Attention is paid especially on low-level statistics methods that are computationally inexpensive compared to learning-based methods. The Grey-World (Buchsbaum, 1980) estimate the color of the light source by estimating the average color in the image assuming that any deviation from average color (Grey) is caused by illuminants. The White-Patch method (Land & others, 1977) estimates the color of light source by computing the maximum response in individual RGB color channels. Finlayson and Trezzi (2004) shows that both Grey-World and White-Patch algorithms are special instantiation of a more general color constancy algorithm based on Minkowski norm called Shades of Grey. Their investigation of best illumination estimation suggests using Minkowski norm, $p = 6$ to obtain optimal performance.

Though we see new method such as the Color Rabbit (Bani?? & Lon??ari??, 2014) which combine multiple local illumination estimations to a global one, these class of methods are more computationally expensive which is not suitable for real-time application. Inspired by primary visual cortex (V1) of human visual system (HVS), Gao, Yang, Li, and Li (2013a) estimate the true illuminant color of a scene by computing the maximum response in separate RGB channels of the responses of double-opponent cells. This method is shown to perform better on outdoor scenes from Gehler-Shi dataset where the mean reflectance is not achromatic which is assumed by Grey-World based methods.

2.2 Saliency Region Detection

Ability of human visual system (HVS) to selectively process only the salient visual stimuli, specifically salient object detection helps to reduce computation time of object recognition that traditionally relies of sliding-window approach to detect object of interest. Achanta, Hemami, Estrada, and Susstrunk (2009) estimate centre-surround contrast using color and luminance features using a frequency-tuned approach to generate high-resolution saliency map. In contrast, biological inspired method of (Itti, Koch, & Niebur, 1998) that computes centre-surround contrast using Difference of Gaussian (DoG) which generates low resolution map and ill-defined boundaries because of down sampling of original image.Because saliency detection often work poorly in low contrast environment i.e underwater environment, work of Van De Weijer and Gevers (2005) boost local color information by analyzing isosalient colour derivatives. Cao and Cheikh (2010) extended work of Van de Weijer as Gaussian derivatives of each opponent color to get a better iso-salient transformation.

2.3 Detection Proposals

Relying on saliency mechanism is insufficient in perturbed underwater condition; therefore, different detection proposals algorithms are investigated. Hosang, Benenson, Dollár, and Schiele (2015) cited that "detection proposals" which can be grouped into a) grouping proposal methods and b) Window scoring proposals methods are used extensively by top performing object detectors in PASCAL and ImageNet. On top of reduced computation cost by avoiding exhaustive sliding window approach, detection proposals improve recall by filtering out false positives. Recent work of Winschel, Lienhart, and Eggert (2016) combines top performing detection proposals methods, SelectiveSearch (Uijlings, van de Sande, Gevers, & Smeulders, 2013) and EdgeBox (Zitnick & Dollár, 2014). Though detection proposals allow for faster object recognition, it is important that it does not filter out object of interest and incur more computation costs that out weights time saved.

2.4 Object Detection and Tracking

An overall review of journal papers submitted by top-performing teams in Robosub shows a general trend of combining surprisingly simple computer vision techniques such as adaptive color thresholding, edge detection i.e Canny Edge (Canny, 1986), and contour analysis i.e Hu moment (Hu, 1962). Team CUAUV (Cornell AUV) proposes adaptive color thresholding on different color spaces such as LAB, LUV and YCrCb where the individual masks are combined to form final binarized mask. This is a blob-based detection approach where contour generated by OpenCV's implementation of (Suzuki & others, 1985) will be matched against known geometric properties of desired object of interest. Walters, Sauder, Nezvadovitz, Voight, Gray, Schwartz, and Walters, P and Sauder, N and Nezvadovitz, J and Voight, F and Gray, A and Schwartz (2014) use particle filter approach to detect and track object of interest. Known for its ability to deal with non-linear noise and multi-modal hypotheses (Isard & Blake, 1998), particle filter has the ability to recover from wrongly tracked objects. Though more sophisticated techniques such as neural-network classification is deployed, teams still generally rely on low-level visual cues such as color and edge. This may be attributed to simplicity and efficiency of mentioned algorithms. Benoit, Goulet, Bouchard-d'Haese, Bouzidi, Carrier, Couturier, Desjardins, Dozois, Fortier, Langlois, Ritchie, and Prévost (2014) focuses on developing sophisticated vision tuning client that allows for rapid prototyping via "mix and match" approach to design a suitable vision pipeline for each individual vision tasks.

Chapter 3

Design & Methodology

3.1 Proposed design

Though many solutions to underwater vision challenges exist, many of them are not designed to work with each other as they do not share a common interface. To increase ease of use and productivity of developers, this paper proposed a vision framework that consists of modular components tailored for underwater application, and ease of integration to Robot Operating System (ROS) which are commonly used by the robotics community.

The proposed vision framework is divided into *offline* modules and *online* modules. *Offline* modules refer to modules that will deployed prior to object tracking mission such as video annotation, visual data analysis and model learning. In contrast, *Online* modules are deployed during mission such as preprocessing, object detection and object tracking.

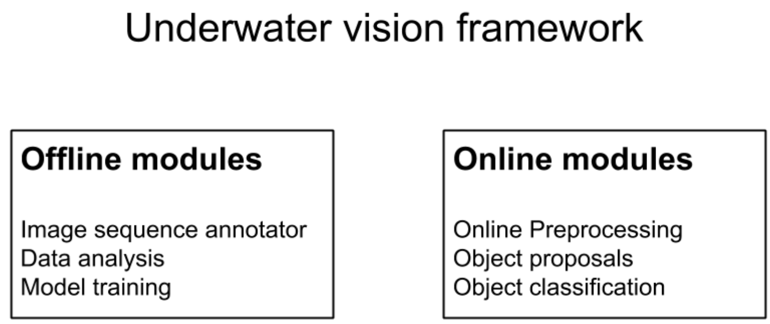


Figure 3.1: Proposed vision framework

3.1.1 Offline modules

Video annotation is extremely important for ground truth generation that is essential for model learning. To achieve rapid ground truth generation with limited manpower and time, model-free tracking method such as mean-shift by (Comaniciu and Meer (2002)) and correlation-filter based tracking (Bolme, Beveridge, Draper, and Lui (2010b)). Of course this is under the assumption that some degree of localization error is acceptable and human intervention is used to redefine the target window if drift occurs. This enables a faster testing iteration as data collected can be integrated more quickly to update our model.

Data analysis helps us discover patterns and statistical nature of collected visual data which is important for feature engineering and model learning. These tools include visualization of image under different color spaces, estimation of illuminants, saliency map generation and image quality assessments. This information is used as metadata to label and categorize dataset to increase productivity of model training and validation. Again, this is an attempt to automate trivial task that require human attention.

Model Training is divided into several stages such as feature selection, model selection and hyperparameters optimization. To increase usability of the software without machine learning knowledge, this paper adopt the trending automatic machine learning approach by leveraging on available open-source libraries such as Auto-Sklearn, TPOT, and HPOlib.

3.1.2 Online modules

Preprocessing has considerable effect on accuracy of underwater object detection because of the challenges mentioned. Color normalization is performed on image to remove effect of color cast because of light attenuation. Low-level stastical methods have been explored because they are simple to implement and fast while producing accuracy comparable to other methods such as gamut mapping and learning methods Gijsenij et al. (2011a). In addition, fusion-based underwater image enhancement by Fang, Deng, Cao, and Fang (2013) is implemented to remove haze effect because of back-scattering of light. Following that, various illumination compensation methods are executed to reduce effect of flickering and adjusting brightness of the image for more optimal object detection.

Object Tracking is separated into 3 components: a) *object proposal*, b) *object classification* and c) *online preprocessing*. An adaptive object model and pre-learned object model are applied to achieve higher tracking accuracy. *Object proposals* based on superpixel, edge-detection and saliency are exploited to produce candidates for classification instead of the traditional sliding-window approach which is more computationally expensive. For *object classification* of candidate windows, Support Vector Machine (SVM), Random Forest and Gaussian Process are the supported classifiers. To improve generalization of the tracker to different conditions, *pre-processing* steps are taken to ensure invariance to non-uniform illuminations and underwater challenges.

3.2 Methodology

In this section we will explain how these modular components are used together for underwater real-time object tracking.

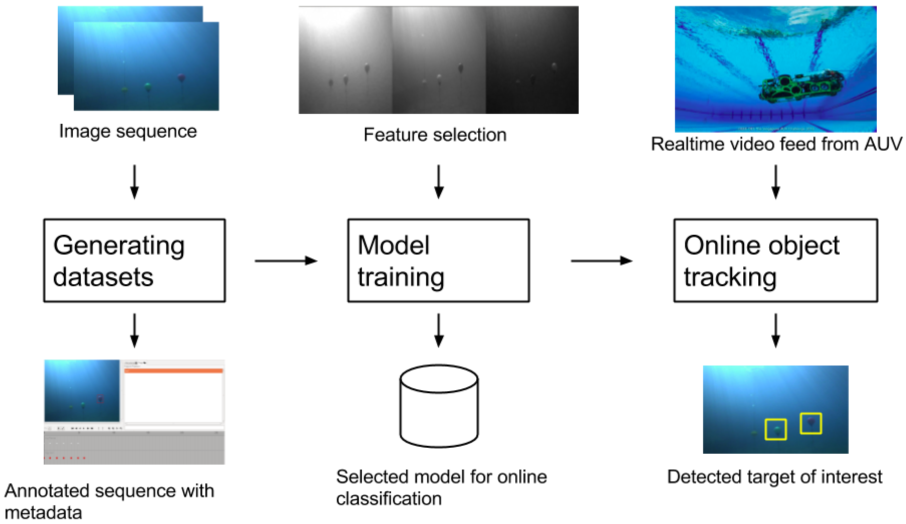


Figure 3.2: Main methodology

3.2.1 Generating datasets

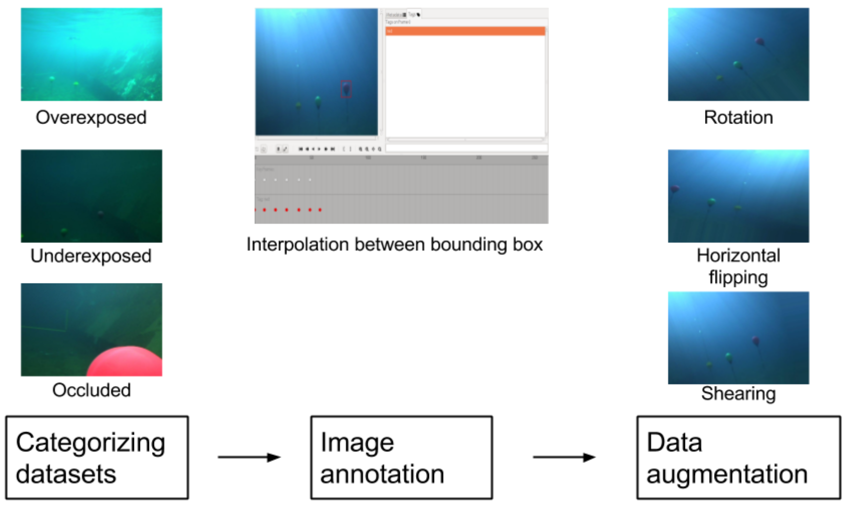


Figure 3.3: Dataset generation methodology

Data analysis is performed on all images to further categorize them into different datasets according to various criteria such degree of haze, existence of shadow, illuminations and color cast. Each dataset will be tagged with metadata generated from the analysis. Next, **video annotation** is conducted using *Mean-Shift* tracker on preprocessed images to generate ground truth that will be used for training and validation. To prevent overfitting and help the model generalize better, data augmentation via horizontal flipping, scaling, rotating, shifting and color jittering is performed with the aid of Keras preprocessing module.

3.2.2 Model Learning

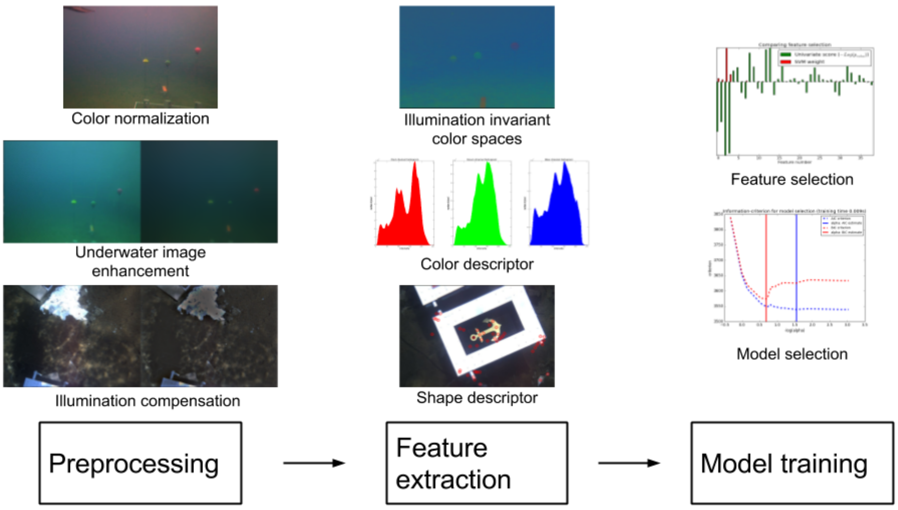


Figure 3.4: Model learning methodology

To improve discriminability of objects from background in undewater setting, **preprocessing** steps are taken such as color normalization, illumination compensation and image enhancement. Moving on, different type of features are extracted from various color spaces that will be used in object classification. Using the validation set, **feature selection**, **model selection** and **hyperparameters optimization** are executed to determine the most optimal combination of algorithm-parameters pair for a particular object class.

3.2.3 Online object detection and tracking

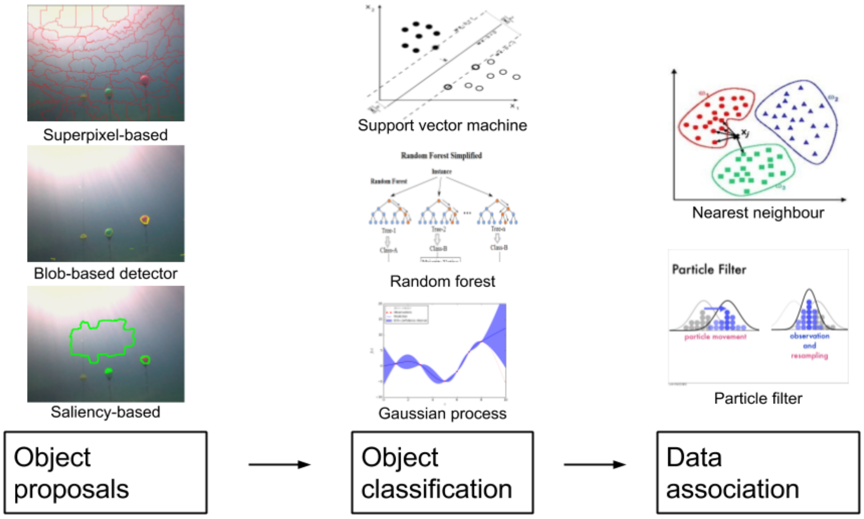


Figure 3.5: Object tracking methodology

The next phase involves real-time **object proposals** adopting a) superpixel-based clustering and b) edge detection. **Object classifier** will rank these candidates according to classfication score. Tracking is performed using a simple nearest neigbour approach and a new tracker will be initialized after losing track of the target for 10 frames.

Chapter 4

Preprocessing

In this section, we will look into detail the preprocessing steps that are applied to each image.

4.1 Color Normalization

For underwater vision challenges, color cues are very important features because other features like edge, texture and corner have poor visibility underwater because of low contrast. In Robosub, there are several vision tasks that requires classification of different objects based on their colors. Though color cue is a simple and discriminative feature for underwater object detection, color feature shows very poor repeatability under varying light source. To achieve consistent feature extraction, this paper takes a *static* approach (fixed-parameters) because of simplicity and our application does not require high degree of accuracy. With the static approach, there are less parameters needed to be optimized.

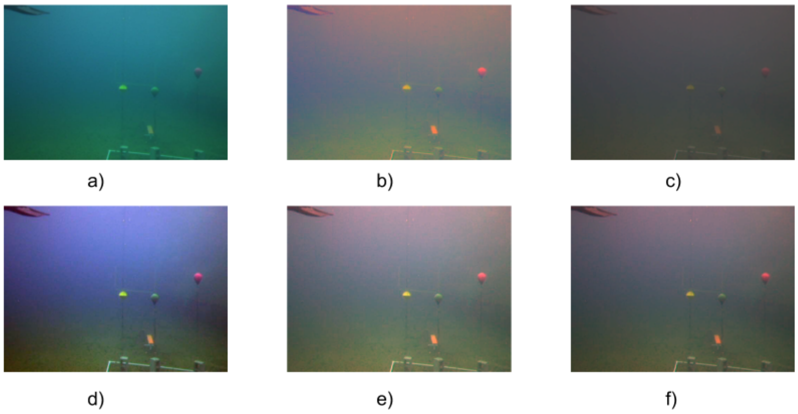


Figure 4.1: Color normalization results (left to right):

Top row: a) Raw input, b) Finlayson's comprehensive normalization, c) Grey-world

Bottom row: d) IACE, e) Finlayson's non-iterative normalization f) Shade of Gray

This paper only considers color normalization methods that require single image without prior information about the camera used. There are 2 main steps to any color normalization method: a) illuminant estimation and b) image correction. The aim of image correction is to achieve chromatic adaptation which can be modelled with a diagonal transformation von Kries (1970) with certain assumptions. The mapping of an image under unknown light source to an image under canonical light source is performed using a diagonal matrix as shown below:

$$\begin{pmatrix} R_c \\ G_c \\ B_c \end{pmatrix} = \begin{pmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & d_3 \end{pmatrix} \begin{pmatrix} R_u \\ G_u \\ B_u \end{pmatrix}$$

Results from Gijsenij, Gevers, and Van De Weijer (2011b) suggests that different algorithms show their strenghts and weaknesses on different datasets. Therefore, this paper proposes a some color normalization strategies that can be chosen based on performance of object detection on the validation datasets.

4.1.1 Algorithm Implementation

Grey-World based

1. **Grey-World**

   With the assumption that: *the average reflectance in a scene under a neutral light source is achromatic* Buchsbaum (1980), the colour of the light source is estimatd by computing the average color in the image.

2. **White patch**

   With the assumption that: *the maximum response of RGB channels is caused by the perfect reflectance* Land and others (1977), the colour of the light source is estimatd by computing the maximum pixel value of each channel separately.

3. **Grey-Edge**

   Instead of using raw pixel value, Van De Weijer and Gevers (2005) makes the assumption that: *the average of the reflectance differences in a scene is achromatic.* The illuminant is estimated by calculating the average color derivative of an image.

4. **Shade of Gray**

   Instead of applying the maximum operation (max RGB) and average operation (Grey-World) which are both specific instantiation using the Minkowski norm Finlayson and Trezzi (2004). Grey-World when $p = 1$ and Max RGB when $p = \infty$.

$$(\frac{\int (|f_x(x)|)^p dx}{\int dx})^{\frac{1}{p}} = ke$$

Finlayson's approach

1. **Comprehensive image normalization**

   To remove dependency on lighting geometry, (r, g, b) is normalized to (s_r, s_g, s_b). Effect of illuminant is removed using grey-world normalization. These two-processes are performed succesively for 2 iterations (derived from empirical results) Finlayson, Schiele, and Crowley (1998).

2. **Non-iterative comprehensive image normalization** Operating on the log RGB space, normalization is performed by subtracting mean of each row and mean of each column each element Finlayson and Xu (2002).

4.1.2 Improvements

1. **Gamma correction**

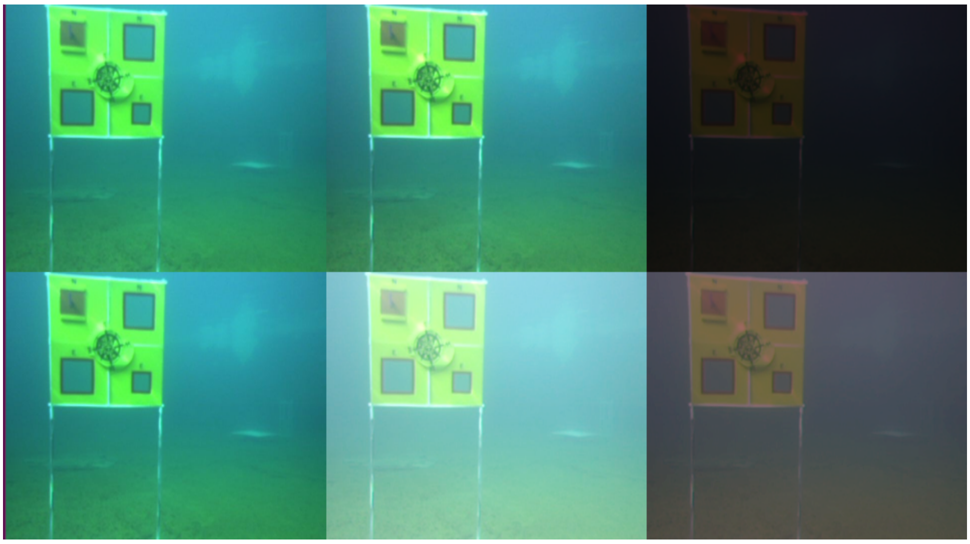


Figure 4.2: Effect of applying gamma correction: (top row) no gamma correction, (bottom row) with gamma correction

To improve the result of color correction Cepeda-Negrete and Sanchez-Yanez (2012), gamma correction is applied after color correction to illuminate dark areas in the image (often effect of color normalization) which subsequently increase dynamic range of the image.

2. **LAB color space**

Based on the evaluation of Kloss (2009), the CIE LAB color space which reflects linearity of human colour perception is able to better represent transformations for more subtle colours. Color normalization in LAB space will rarely overcompensate or result in transformed image that looks unnatural.

3. **Grey pixel**

Figure 4.3: Applying novel grey pixel illumination estimation: a) Raw input, b) Color corrected

Yang, Gao, and Li (2015) estimates the illuminant of the scene from information of grey pixels detected in a color image. It assumes that *most of the natural images include some detectable pixels that are at least approximately grey.*Firstly, color image is converted to logarithm space, followed by calculating the illumination-invariant measure (IIM) which is calculated from local contrast of each logarithm channels. Then the mean of selected grey pixels ranked by the Grey-Index will give us the estimated illumination.

4. **PCA based**

Figure 4.4: Spatial domain based illumination estimation: a) Raw input, b) Color corrected

The work of Cheng, Prasad, and Brown (2014a) estimates the illuminant by finding bright pixels and dark pixels in a color image. The paper selects colours by choosing n pixels with largest and smallest projected distance to the mean vector. Then PCA is performed on the selected pixels to generate estimated illumination direction.

4.2 Underwater Image Enhancement

Color normalization alone is insufficient to restore the original appearance of an underwater obstacles. Underwater images also suffer from poor contrast, overexposed or underexposed and flickering caused by refraction of sun light.

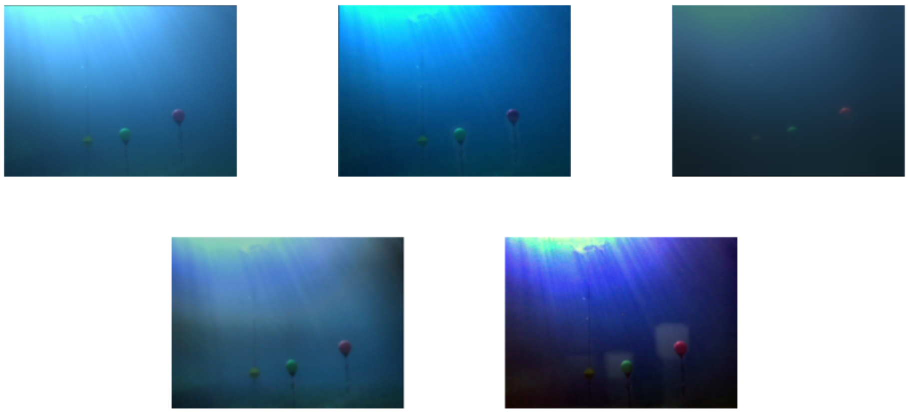


Figure 4.5: Underwater image enhancement results (left to right):

Top row: a) Raw input, b) Dark channel prior, c) Single image fusion

Bottom row: d) CLAHE, e) Red channel prior

4.2.1 Fusion-based image restoration

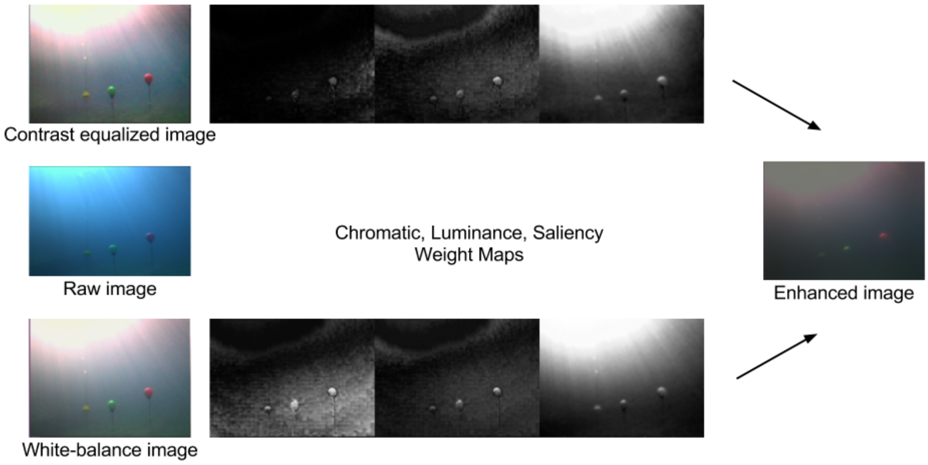


Figure 4.6: Single underwater image enhancement by fusion

The work of Fang et al. (2013) suggests enhancement of underwater images using a) white-balance image and b) contrast equalized image as inputs to generate weight maps (chromatic, luminance, and saliency). The weight maps are then normalized and fused using image pyramid approach to produce a smoother enhanced image. The final output can be gamma corrected to adjust overall brightness of the image.

Chromatic map controls the saturation gain of the enhanced image. Higher saturation values yield more vivid color. **Luminance map** helps to balance the brightness of the enhanced image while **Saliency map** indicates area of high conspicuity. In other words, saliency map higlights area that captures attention of the human visual system.

This approach has the added benefit of being extremely computationally fast and simple to implement compared to other approach such as dark channel prior He, Sun, and Tang (2011).

4.2.2 Denoising & Illumination Compensation

According to the survey by Padmavathi, Subashini, Kumar, and Thakur (2010), filters like homomorphic filter, anisotropic filter and wavelet denoising filter are necessary to suppress noise, preserve edge and smoothen underwater image. The vision framework includes the following filters:

Figure 4.7: Illumination compensation results (left to right):

Top row: a) Underexposed input, b) Chih's light compensation, c) Chen's light compensation

Bottom row: d) Flicker input, e) Homomorphic filter f) Gamma corrected

1. **Homomorphic filter**

   Underwater vision tasks in shallow water are prone to suffer from spatial temporal illumination patterns. In this case, a homomorphic filter can help to correct non-uniform illumination and sharpen the image. With the assumption that the high frequency components of an image is associated with reflectance of the image, a high pass filter is applied on the frequency domain (removing multiplicative noise) removing the low frequency (flickers).

2. **Anisotropic filter**

   Use in conjuction with the homomorphic filter is the anisotropic filter which smoothens homogeneous area while preserving edges. The work of Perona and Malik (1990) helps to reduce small edges generated by homomorphic filter.

3. **Illumination compensation**

   When executing underwater vision tasks, the AUV has to constantly deal with fluctuation in illumination because of various factors such as position of the sun and clouds. Instead of relying on manual tuning of camera parameters, some automated light compensation is performed on captured image sequence. This is extremely important as an overexposed or underexposed image lose most of its chromatic information.

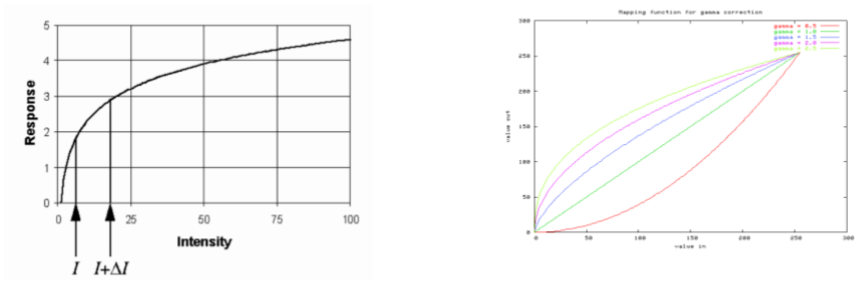


Figure 4.8: Comparison between logarithm curve and gamma curve

This paper refers to the work of Chang () for 2 different brightness adjustment algorithms. Firstly, a logarithm curve is used which obeys the Weber-Fechner law of JND (Just noticeable difference) response instead of a gamma curve which tends to enhance noise in dark regions.

4.3 Conclusion

The preprocessing stage is one of the most important component for effective underwater object tracking. Color normalization ensures repeatability of feature extraction for different datasets. This allow for extracting domain invariant features which can be used for object tracking in different water environments such as public swimming pool, beach or a man-made lake (venue of Robosub competition).

However, it is necessary to keep the selection of preprocessing algorithms small to reduce any overhead on real-time object tracking. Therefore, this paper favors preprocessing algorithms that are less complex, effectively trading off some degree of accuracy for lower detection latency.

Chapter 5

Object Proposals

Recently we have seen more state-of-the-art trackers incorpate object proposals as part of their pipeline Kristan, Leonardis, Matas, Felsberg, Pflugfelder, Čehovin, Vojir, Häger, Lukežič, and Fernandez (2016), Kristan, Matas, Leonardis, Felsberg, Cehovin, Fernandez, Vojir, Hager, Nebehay, and Pflugfelder (2015). The rise of object proposals which is a segmentation-based candidates generation slowly replaces the more traditional sliding window approach which can be slow when multiscale detection is required.

In addition, object proposals can be thought as a generic object detector which generate candidate window based on some measure of objectness. Choosing the criteria to measure the presence of an object is very important and is unique for each domain of application. Object proposals according to Hosang, Benenson, Dollár, and Schiele (2016) fall into 2 large categories: a) grouping and b) window ranking.

Grouping proposals leverage on hierarchical segmentation approach to generate overlapping segments with techniques such as a) superpixel grouping (SP), b) solving multiple graph cut problem (GC) with random seeds or from c) edge contour (EC). On the other hand, **Window scoring** proposals only score each candidate window on likelihood of containing an object. This approach is faster at the cost of lower localization accuracy.

From the mentioned paper, methods that are based on superpixel are not robust towards illuminatin change while **BING** Cheng, Zhang, Lin, and Torr (2014b) and **Edge-box** Zitnick and Dollár (2014) show promising result because of its machine learning component (random forest).

5.1 Algorithm Implementation

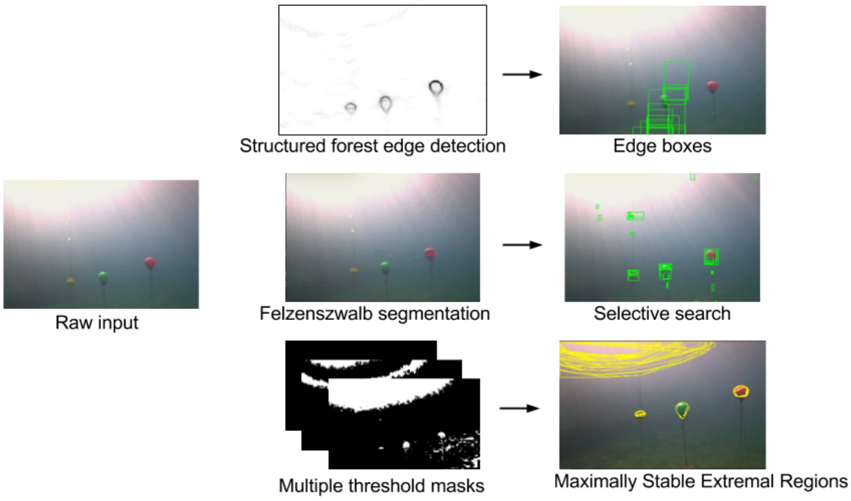


Figure 5.1: Different object proposal paradigms

This paper utilizes 4 different object proposals for different type of underwater vision tasks. Referring to Figure 5.1, detection of color buoys which are largely homogeneous with little edge information are much better handled with **SelectiveSearch** Uijlings et al. (2013) approach. In general, *grouping proposals* show more promising result than *window scoring* approach for all the underwater vision tasks. The additional speed gained from *window scoring* approach is almost nullified with the need to sample large amount of windows to achieve decend localiztion accuracy.

Besides the *SelectiveSearch* and *Edge-box*, the following section will discuss custom implementation of 2 other proposal methods.

5.1.1 Maximially Stable Extremal Regions (MSER)

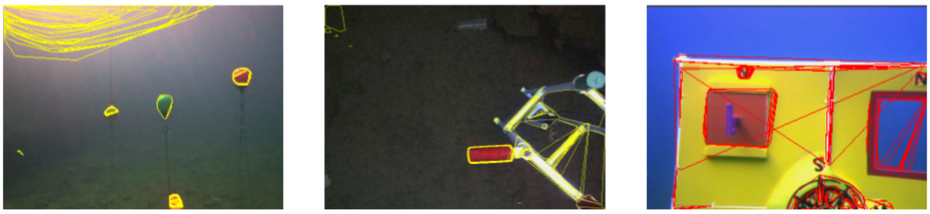
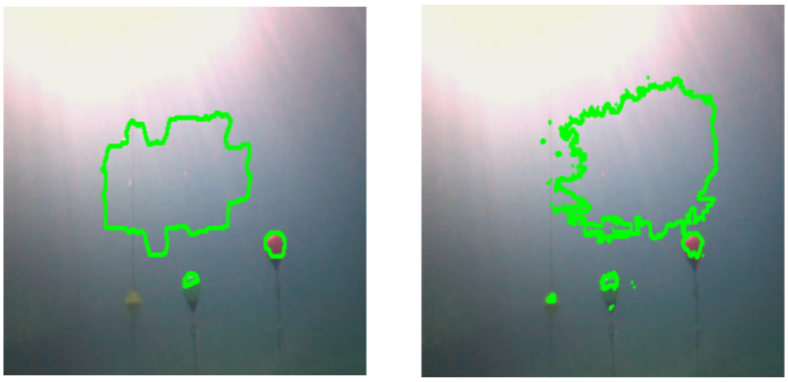


Figure 5.2: Object proposals using MSER: a) Buoy task, b) Coin task, c) Set date task

Since most the underwater obstacles are blob-like , this paper uses the implementation of Forssén (2007) by OpenCV to extract candidate windows for object detection. Firstly, the image is converted to HSV color space. The *Saturation* channel is then used for blob-detection as most underwater obstacles have more vivid color compared to the background. Alternatively, a combination of different color channels are explored to generate more segments such as L*a*b and YUV.

5.1.2 Saliency-based

Object proposals based on salient cues are also explored as they mimic closely how human visual system works. Without any preprocessing, the results of salient object proposal is mediocre at best compared to the other proposal methods. However with appropriate color normalization and enhancement, this method can produce results that can rival with *SelectiveSearch* and *Edge-box* in this domain of application. This paper use an open-source implementation of:



Salient optimization using robust background subtraction

Minimum barrier salient object detection at 80 FPS

Figure 5.3: Object proposals using saliency approach

1. **Saliency optimization from robust background subtraction** Zhu, Liang, Wei, and Sun (2014)

2. **Minimum Barrier Salient Object Detection at 80 FPS** Zhang, Sclaroff, Lin, Shen, Price, and Mech (2015)

3. **Frequency-tuned Salient Region Detection** Achanta et al. (2009)

5.2 Conclusion

In this section we have explored different proposal methods that are state-of-the-art and others (MSER and saliency) that are slightly different from available literatures. Though *MSER* managed to generate a lot of segments, this method does not measure the quality of each segment unlike *Edge-box*. *Salient object proposals* on the hand produce candidate windows that are highly accuracte but with very few segments. In general, not missing any possible object candidate is more important than generating highly accurate candidate window. Therefore, this paper propose to lower the threshold for *saliency-based* proposals in order to generate more candidate windows.

Chapter 6

Feature Design

Large amount of feature detector and descriptors used for this projects are available in OpenCV or Scikit-image. This paper has come to this list of features based on the benchmarks by Lee, Jeon, Yoon, and Paik (2016) and Pieropan, Björkman, Bergström, and Kragic (2016). Below is a summary of features available in this vision framework:

1. **SURF** Bay, Tuytelaars, and Van Gool (2006)
2. **SIFT** Lowe (1999)
3. **BRISK** Leutenegger, Chli, and Siegwart (2011)
4. **ORB** Rublee, Rabaud, Konolige, and Bradski (2011)
5. **FREAK** Alahi, Ortiz, and Vandergheynst (2012)
6. **MSER** Forssén (2007)
7. **DAISY** Tola, Lepetit, and Fua (2010)
8. **CenSure** Agrawal, Konolige, and Blas (2008)
9. **LBP** Ojala, Pietikainen, and Maenpaa (2002)
10. **AKAZE** Alcantarilla and Solutions (2011)
11. **Inner Shape Context** Ling and Jacobs (2007)

12. **Elliptic Fourier Feature of Closed Contour** Kuhl and Giardina (1982)

13. **Histogram of Oriented Gradient** Dalal and Triggs (2005)

14. **Hu moment and Zernike moment** Sabhara, Lee, and Lim (2013)

6.1 Requirements

There are various desired properties of features for underwater object tracking in particular ones that are highly applicable in the competition setting of Robosub competition.The 2 main properties are: a) *repeatability* and b) *discriminability.* Firstly, it is important that we are able to consistently extract the same set of features for the same object in order to achieve consistent detection. However, there is always a trade-off with *discriminability* as features that are highly repeatable tend to describe a more general representation of the object.

6.1.1 Illumination invariance

It is preferable to have features that are highly invariant to sudden changes in illumination as the operational depth of the AUV during the competition is still susceptible to external factors such as position of the sun and clouds.To achieve this, this paper propose usafe of multiple illumination invariation color space to describe appearance of the object.

6.1.2 Scale & Rotation invariance

Secondly, the feature must also be scale and rotation invariant because the AUV will be constantly navigating around its surrounding to identify object of interest. The easies way to achieve this is to rely on purely color-based feature that will be mentioned in section 6.2. Color-based features are easier to compute and are less computationally expensive compared to features like SURF, SIFT and HOG.

6.1.3 Shape discriminability

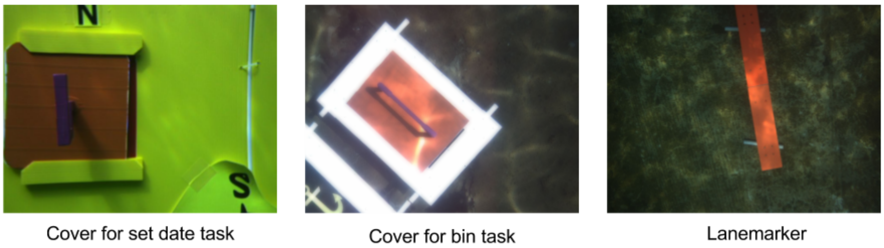


Figure 6.1: Objects with similar colors

Color-based features alone are not sufficient for our application as there exists objects of similar color appearance. In this case, we will need shape descriptors that will be elaborated in section 6.3.

6.2 Color space: Implementation

Besides the usual color spaces, this section will look at some implementation of color spaces based on these papers on the subject:

1. Detecting salient cue through color-ratio Todt and Torras (2004)
2. Color Invariants for Person Re-identification Kviatkovsky, Adam, and Rivlin (2013)
3. Evaluation of Color Descriptors for Object and Scene Recognition Van De Sande, Gevers, and Snoek (2010)
4. Invariant color descriptors for efficient object recognition Sande and others (2011)
5. A Perception-based Color Space for Illumination-invariant Image Processing Chong, Gortler, and Zickler (2008)

6. Illumination invariant color model robot soccer Luan, Qi, Song, Chen, Zhu, and Wang (2010)

7. Illumination invariant imaging Maddern, Stewart, McManus, Upcroft, Churchill, and Newman (2014)

8. Color Model Double Opponency Gao, Yang, Li, and Li (2013b)

6.2.1 rg chromacity

$$r = \frac{R}{R+G+B}, g = \frac{G}{R+G+B}, b = \frac{B}{R+G+B}$$

6.2.2 Normalized RGB

$$r = \frac{R-\mu(R)}{\sigma(R)}, g = \frac{G-\mu(G)}{\sigma(G)}, b = \frac{B-\mu(B)}{\sigma(B)}$$

6.2.3 Opponent color space

$$O1 = \frac{R-G}{\sqrt{2}}, O2 = \frac{R+G-2B}{\sqrt{6}}, O3 = \frac{R+G+B}{\sqrt{3}}$$

$$W_{o1} = \frac{O1}{O3}, W_{o2} = \frac{O2}{O3}$$

6.2.4 Log color ratio

$$L1 = \log\frac{R}{B}, L2 = \log\frac{R}{B}, L3 = \log\frac{G}{B}$$

6.2.5 RGBY opponent space

$$R_o = R - \frac{G+B}{2}, G_o = G - \frac{R+B}{2}, B_o = B - \frac{R+G}{2}, Y_o = \frac{R+G}{2} - |R-G| - B$$

6.2.6 DCD: Dominant color descriptor

Convert to LUV color space or any other perceptually unifrom color space. Perform K-mean clustering and return the percentage of pixels and variance of each color centers.

6.3 Shape descriptors

6.3.1 Inner shape context

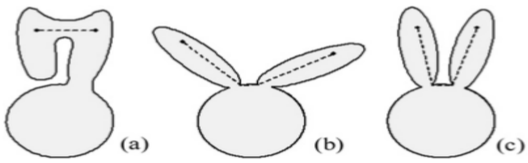


Figure 6.2: Dashed lines denote shortest path withint the shape boundary

Inner-distance is defined as the length of the shortest path between landmark points within the shape silhouette. Inner distance is used instead of Euclidean distance when building the shape context. This is an improvement over the traditional shape context as it is able to describe more complicated shapes.

6.3.2 Elliptic Fourier Feature of Closed Contour

A chain-encoded closed contour is first obtained using OpenCV's *findContour*. Normalization of Fourier's coefficients using various elliptical properties of the coefficients. This descriptors obtained are invariant to *rotation*, *dilation* and *translation* of the contour.

6.3.3 Moment-based descriptors

The vision framework includes 2 common moment-based descriptors: a) *Hu Moment* and b) *Zernike Moment*. According to the evaluation by Sabhara et al. (2013), Zernike's moment is more robust and flexible as one can varies the order of polynomial to describe more complex shape. Furthermore, the Pseudo-Zernike's moment which more robust to noise is also part of the supported shape descriptors.

In addition, simpler contour properties can also be used if the target of interests consist of basic shapes that are largley different from each other. These properties include:

1. **Eccentricity**

   Fit a bounding box over the closed contour to obtain the lenght of major axis and length of the minor axis.

$$Eccentricity = \frac{L_{majoraxis}}{L_{minoraxis}}$$

2. **Circularity ratio**

   Calculates the ratio between the area of original shape and area of its enclosing circle.

$$Circularity = \frac{Area_{shape}}{Area_{circle}}$$

3. **Rectangularity**

   Similar to above, this calculates the ratio between area of the original shape and area of its enclosing rectangle.

$$Rectangularity = \frac{Area_{shape}}{Area_{rectangle}}$$

Chapter 7

Model Learning

The primary classifiers for object classification include: a) *SVM*, b) *Random Forest*, and *Gaussian Process*. These classifiers are selected primarily because we have small amount of data as availability of undewater data are quite limited and can be very expensive to collect. Neural network and its more popular sibling: Deep Neural Network is largely ignored because of a) scarcity of data and b) many parameters tuning are needed. In addition, this vision framework hopes to achieve comparable accuracy in underwater object tracking relying more simple features and less parameters intensive tuning from human experts.

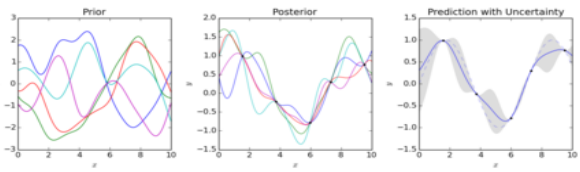


Figure 7.1: Gaussian process

Gaussian process (GP) Rasmussen (2006) is introduced because of its unique ability to perform feature selection using a covariance function that implements *automatic relevance determination*. A GP model also provides uncertainty scores of each classification and a prior knowledge can be integrated easily into its prior function. One major downside of GP is definitely its $O(n^3)$ complexity which makes it a poor choice for large amount of data.

The following section will focus more on the effort to apply the *automatic machine learning* principle which aims to automate trivial machine learning tasks such as *feature selection*, *model selection* and *hyperparameters optimization.* Most of these algorithms are open-source and are readily available, this paper merely tries to integrate it as part of the framework to remove the dependency on machine learning experts for trivial tasks.

7.1 Feature Selection

Having multiple features is advantageous to allow for greater adaptation to different challenging environments. For instance, detection of a textureless object can be challenging using the popular HOG feature while a simple color histogram can produce a better result. This paper would like to highlight that choosing the right feature can improve accuracy and reduce needless computational cost from using complex feature descriptors like SIFT. In addition, choosing best features also reduce dimension of feature which is a problem on small AUV equipped with less powerful computing unit.

Besides using the *Automatic relevance determination* of a GP model, this framework leverages on the widely used machine learning library, *Sklearn's feature selection module.* Below is the list of feature selection functions used in the vision framework.

1. **Removing feature with low variance**

   This approach removes features that are below certain variance threshold labelling it as redundant.

2. **Univariate feature selection** Univariate test such as F-test and chi-test are used to select the best features before training the model.

3. **Tree-based** Uses an ensemble model (forest of trees) to calculate feature importances which is used to score input features.

7.2 Hyperparameters optimization: Implemetation

Uses HPOlib which contains 3 libraries for hyperparameter optimizations:

1. Sequential model-based optimization Hutter, Hoos, and Leyton-Brown (2011)
2. Spearmint Bayesian optimization codebase Snoek, Larochelle, and Adams (2012), Swersky, Snoek, and Adams (2013), Snoek, Swersky, Zemel, and Adams (2014), Snoek (2013), Gelbart, Snoek, and Adams (2014)
3. Hyperopt

Ideally, there are very few hyperparameters optimization needed as the paper actively tries to use non-parametric methods with the exception of SVM (choice of kernel and misclassificatin penalty).

7.3 Model Selection

For model selection, again *Sklearn's model selection and evaluation* is used to select the best model for a particular tasks through cross-validation. From our observation, the performance of each classifiers does not varies very much. In addition, this paper also experimented with *Auto sklearn* Feurer, Klein, Eggensperger, Springenberg, Blum, and Hutter (2015).

Chapter 8

Object Tracking

This section will explore different tracking paradigm Stalder, van Gool, and Avidan (2012) and analyze various surveys conducted to determine the best trackers for underwater visual tracking. Because of the tracking strategy, only single object tracking algorithms are evaluated. There are few reasons why the paper proposed a single object tracking approach:

1. Simpler implementation
2. The AUV has limited number of manipulators which make it possible of manipulating only a single obstacle at a time
3. Less computationally expensive

8.1 Benchmarks

As for the benchmarks, the paper looks into the papers listed below:

1. Visual object tracking performance measures revisited Čehovin, Leonardis, and Kristan (2016)
2. VOT 2016 Challenge Results Kristan et al. (2016)
3. VOT 2015 Challenge Results Kristan et al. (2015)
4. Is my new tracker really better than yours ? Čehovin, Kristan, and Leonardis (2014)

The top trackers almost always combine an adaptive tracking and a fixed-model tracking approach. Online model update techniques such as *Adaboost* and *Multiple Instance Learning* are capable of adapting to different conditions as positive samples are sampled around vicinity of tracked object while negative samples are extracted from background of the image. However, adaptive model update comes at the cost of computational cycle and also more complicated model.

8.2 Tracking by detection: Implementation

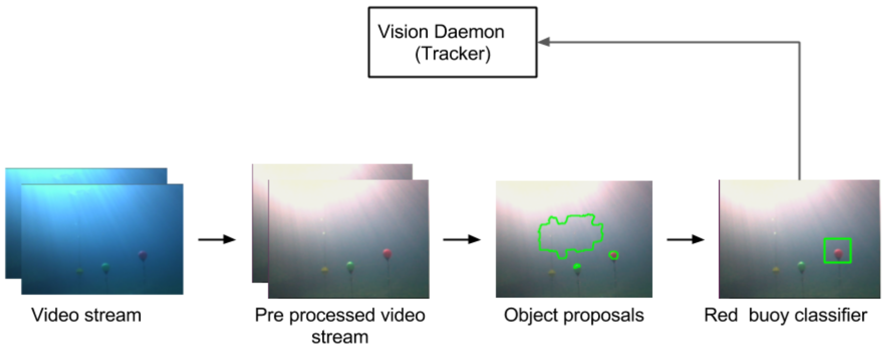


Figure 8.1: Tracking pipeline

This paper utilizes a tracking by detection approach as detection is performed on each frame and associated with previously tracked objects. A tracker is **terminated** when an the tracker loses track of its target for at least 10 frames. This value is determined through empirical evaluation of applying the tracker on existing image sequence. To handle **multiple instance** of the object, the object with the shortest Euclidean distance will be selected. In addition, association of object with previously tracked object is bounded on a specific radius to reduce false positives with the assumption that the AUV is perfectly stable and does not move randomly over a short period of time.

Prior knowledge such as geometric property of the target can be included through a weighted summation of classification score and prior score. With more context, a more accurate detection can be achieved.

8.3 Model-free tracking: Implementation

In addition to the main tracking strategy mentioned above, model-free tracking algorithms based on correlation-filter are also included for: a) rapid data collection and b) tracking generic object. These algorithms include:

1. High-speed kernelized correlation filter Henriques, Caseiro, Martins, and Batista (2015)
2. Visual Object Tracking using Adaptive Correlation Filters Bolme, Beveridge, Draper, and Lui (2010a)

Chapter 9

Experimental results

9.1 Datasets

The datasets are generated and categorized using the sequence annotator, AIBU which is used by the Visual Object Tracking (VOT) committee. There are total of 6 datasets with different set of challenges. At the same time, 6 object classes will be tested.

9.1.1 Challenges

Figure 9.1 and 9.2 are the datasets labelled with bounding box ground truth used for evaluation of the proposed tracker.

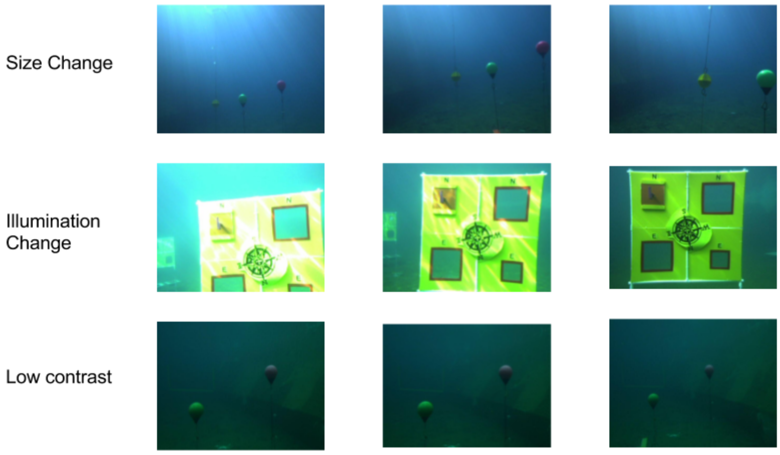


Figure 9.1: Dataset 1

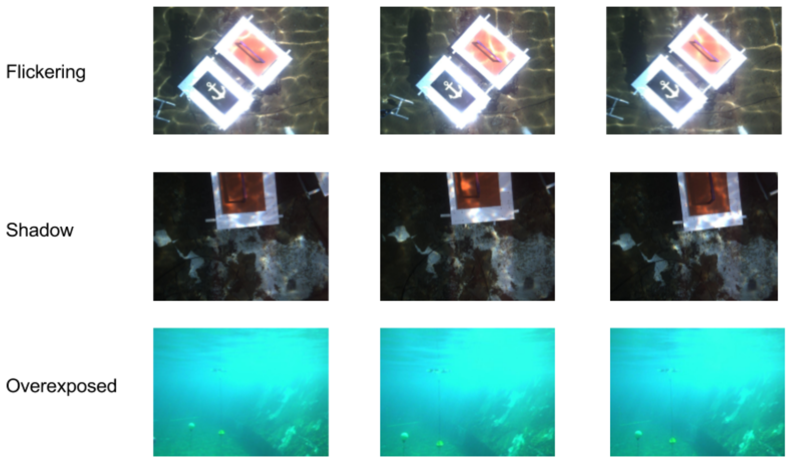


Figure 9.2: Dataset 2

9.1.2 Object classes

The objects to be tracked composed of:

1. Red buoy
2. Green buoy
3. Yellow buoy

4. Set date cover

5. Red Coin

6. Bin cover

9.2 Results

9.2.1 Evaluation methodology

Using Visual Object Tracking (VOT15) Kristan et al. (2015) as guideline, following are the performance measures used:

1. **Accuracy, A**

   Accuracy is measured the average overlap of predicted bounding box with the ground truth bounding box. Accuracy for a sequence is obtained by averaging per-frame accuracies.

2. **Robustness, R**

   Robustness measures how many times the tracker loses the target (overlap is zero). The tracker is reinitialized 10 frames after the failure. Again, robustness of a sequence is calculated using the average failure rate.

3. **Frame per-second, FPS** This is a naive measure of speed by calculating the average of FPS of the tracker over different datasets.

9.2.2 Trackers

The competing trackers can be categorized into 2 big categories: a) variations of proposed tracker and b) open-source trackers. The baseline tracker is our proposed tracker without any preprocessing, using only color thresholding along with contour properties. Below is the list of trackers:

Trackers
Baseline
Baseline + preprocessing
Baseline + preprocessing + automl
MOSSE
KCF
EBT (Edge Box Tracker)

Table 9.1: Competing trackers

It is to be made known that only minimal preprocessing such as smoothing and denoising are performed when using open-source trackers. This is to ensure that the inputs are not too perturbed with noise. This makes sure that the state-of-the-art trackers are not at a big disadvantage compared to our proposed trackers.

9.2.3 Raw results

Trackers	Accuracy	Robustness	Speed
Baseline	0.21	7.23	200
Baseline + preprocessing	0.34	6.11	50
Baseline + preprocessing + automl	0.53	2.53	50
MOSSE	0.30	8.10	100
KCF	0.35	4.91	70
EBT (Edge Box Tracker)	0.41	3.11	43

Table 9.2: Raw results across all datasets

9.3 Discussion

9.3.1 Preprocessing

Both correlation filter based trackers, KCF and MOSSE performed poorly for the *illumination-dataset* while our proposed tracker managed to consistently track the object of interest. EBT on the other hand showed poor performance in both *overexposed* and *low contrast* datasets as the edge information is barely visible. For the *size change* dataset, KCF and EBT in particular shows the best performance. This is to be expected as these trackers did show promosing result in VOT15. The baseline tracker without any preprocessing performed miserably in almost all datasets. However, with added preprocessing, the basline tracker is able to achieve decent accuracies for datasets without any complex shapes.

9.3.2 Automatic machine learning

The result for performing feature selection showed promising result as it is able to perform up to par with some of the state of the art trackers such as EBT and KCF. However, it has to be mentioned that these trackers with preprocessing are able to outperform the proposed tracker. This goes to show the importance of preprocessing when performing object detection in underwater environment.

9.3.3 Conclusion

Looking at the Table 9.2, one can conclude the importance of preprocessing for underwater object tracking because the accuracies achieved by state-of-the-art trackers do not justify their ranking in VOT15. Our proposed tracker which combines both preprocessing and automatic machine learning approach is able improve the baseline accuracies by leaps and bounds without needing to really complex feature representations.

References

Achanta, R., Hemami, S., Estrada, F., & Susstrunk, S. (2009). Frequency-tuned salient region detection. *Computer vision and pattern recognition, 2009. cvpr 2009. ieee conference on* (pp. 1597–1604), IEEE, 2009.

Agrawal, M., Konolige, K., & Blas, M. R. (2008). Censure: Center surround extremas for realtime feature detection and matching. *European Conference on Computer Vision* (pp. 102–115), Springer, 2008.

Alahi, A., Ortiz, R., & Vandergheynst, P. (2012). Freak: Fast retina keypoint. *Computer vision and pattern recognition (CVPR), 2012 IEEE conference on* (pp. 510–517), Ieee, 2012.

Alcantarilla, P. F., & Solutions, T. (2011). Fast explicit diffusion for accelerated features in nonlinear scale spaces. *IEEE Trans. Patt. Anal. Mach. Intell*, *34*(7), 2011, 1281–1298.

Ancuti, C., & Bekaert, P. (2011). Enhancing Underwater Images by Fusion. *Computer*, , 2011, 4503–4503.

Bani??, N., & Lon??ari??, S. (2014). Color rabbit: Guiding the distance of local maximums in illumination estimation. *International Conference on Digital Signal Processing, DSP*, *2014-Janua*(Cd), 2014, 345–350.

Bay, H., Tuytelaars, T., & Van Gool, L. (2006). Surf: Speeded up robust features. *Computer vision–ECCV 2006*, , 2006, 404–417.

Benoit, M., Goulet, M.-A. B., Bouchard-d'Haese, F., Bouzidi, R., Carrier, J., Couturier, É., Desjardins, V., Dozois, A., Fortier, M., Langlois, F., Ritchie, K., & Prévost, J. S.-J. (2014). S.O.N.I.A - Autonomous Underwater Vehicle. , 2014.

Bolme, D., Beveridge, J. R., Draper, B. a., & Lui, Y. M. (2010a). Visual object tracking using adaptive correlation filters. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, , 2010, 2544–2550.

Bolme, D. S., Beveridge, J. R., Draper, B. A., & Lui, Y. M. (2010b). Visual object tracking using adaptive correlation filters. *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on* (pp. 2544–2550), IEEE, 2010.

Buchsbaum, G. (1980). A spatial processor model for object colour perception. *Journal of the Franklin institute*, *310*(1), 1980, 1–26.

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6), 1986, 679–698.

Cao, G., & Cheikh, F. A. (2010). Salient region detection with opponent color boosting. *2010 2nd European Workshop on Visual Information Processing, EUVIP2010*, , 2010, 13–18.

Cepeda-Negrete, J., & Sanchez-Yanez, R. E. (2012). Combining color constancy and gamma correction for image enhancement. *Electronics, Robotics and Automotive Mechanics Conference (CERMA), 2012 IEEE Ninth* (pp. 25–30), IEEE, 2012.

Chang, C.-F. Light compensation, .

Cheng, D., Prasad, D. K., & Brown, M. S. (2014a). Illuminant estimation for color constancy: why spatial-domain methods work and the role of the color distribution. *JOSA A*, *31*(5), 2014, 1049–1058.

Cheng, M.-M., Zhang, Z., Lin, W.-Y., & Torr, P. (2014b). Bing: Binarized normed gradients for objectness estimation at 300fps. *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3286–3293), 2014.

Chong, H. Y., Gortler, S. J., & Zickler, T. (2008). A perception-based color space for illumination-invariant image processing. *ACM Transactions on Graphics (TOG)*, Vol. 27 (p. 61), ACM, 2008.

Comaniciu, D., & Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, *24*(5), 2002, 603–619.

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, Vol. 1 (pp. 886–893), IEEE, 2005.

Fang, S., Deng, R., Cao, Y., & Fang, C. (2013). Effective single underwater image enhancement by fusion. *JCP*, *8*(4), 2013, 904–911.

Feurer, M., Klein, A., Eggensperger, K., Springenberg, J., Blum, M., & Hutter, F. (2015). Efficient and robust automated machine learning. *Advances in Neural Information Processing Systems* (pp. 2962–2970), 2015.

Finlayson, G., Schiele, B., & Crowley, J. (1998). Comprehensive colour image normalization. *Computer VisionECCV'98*, , 1998, 475–490.

Finlayson, G., & Xu, R. (2002). Non-iterative comprehensive normalisation. *Conference on Colour in Graphics, Imaging, and Vision*, Vol. 2002 (pp. 159–163), Society for Imaging Science and Technology, 2002.

Finlayson, G. D., & Trezzi, E. (2004). Shades of gray and colour constancy. *Color and Imaging Conference*, Vol. 2004 (pp. 37–41), Society for Imaging Science and Technology, 2004.

Forssén, P.-E. (2007). Maximally stable colour regions for recognition and matching. *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on* (pp. 1–8), IEEE, 2007.

Galdran, A., Pardo, D., Picón, A., & Alvarez-Gila, A. (2015). Automatic Red-Channel underwater image restoration. *Journal of Visual Communication and Image Representation*, *26*, 2015, 132–145.

Gao, S., Yang, K., Li, C., & Li, Y. (2013a). A Color Constancy Model with Double-Opponency Mechanisms. *Computer Vision (ICCV), 2013 IEEE International Conference on*, , 2013, 929–936.

Gao, S., Yang, K., Li, C., & Li, Y. (2013b). A color constancy model with double-opponency mechanisms. *Proceedings of the IEEE International Conference on Computer Vision* (pp. 929–936), 2013.

Garcia, R., Nicosevici, T., & Cufí, X. (2002). On the way to solve lighting problems in underwater imaging. *OCEANS'02 MTS/IEEE*, Vol. 2 (pp. 1018–1024), IEEE, 2002.

Gelbart, M. A., Snoek, J., & Adams, R. P. (2014). Bayesian optimization with unknown constraints. *arXiv preprint arXiv:1403.5607*, , 2014.

Gevers, T., Gijsenij, A., Van de Weijer, J., & Geusebroek, J.-M. (2012). *Color in computer vision: fundamentals and applications*, Vol. 23. John Wiley & Sons.

Gijsenij, A., Gevers, T., & Van De Weijer, J. (2011a). Computational color constancy: Survey and experiments. *IEEE Transactions on Image Processing*, *20*(9), 2011, 2475–2489.

Gijsenij, A., Gevers, T., & Van De Weijer, J. (2011b). Computational color constancy: Survey and experiments. *IEEE Transactions on Image Processing*, *20*(9), 2011, 2475–2489.

Gracias, N., Negahdaripour, S., Neumann, L., Prados, R., & Garcia, R. (2008). A motion compensated filtering approach to remove sunlight flicker in shallow water images. *Oceans 2008*, , 2008, 1–7.

He, K., Sun, J., & Tang, X. (2011). Single image haze removal using dark channel prior. *IEEE transactions on pattern analysis and machine intelligence*, *33*(12), 2011, 2341–2353.

Henriques, J. F., Caseiro, R., Martins, P., & Batista, J. (2015). High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*(3), 2015, 583–596.

Hosang, J., Benenson, R., Dollár, P., & Schiele, B. (2015). What makes for effective detection proposals? *Arxiv*, , 2015, 2014.

Hosang, J., Benenson, R., Dollár, P., & Schiele, B. (2016). What makes for effective detection proposals? *IEEE transactions on pattern analysis and machine intelligence*, *38*(4), 2016, 814–830.

Hu, M.-K. (1962). Visual pattern recognition by moment invariants. *IRE transactions on information theory*, *8*(2), 1962, 179–187.

Hutter, F., Hoos, H. H., & Leyton-Brown, K. (2011). Sequential model-based optimization for general algorithm configuration. *International Conference on Learning and Intelligent Optimization* (pp. 507–523), Springer, 2011.

Isard, M., & Blake, A. (1998). Condensationconditional density propagation for visual tracking. *International journal of computer vision*, *29*(1), 1998, 5–28.

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

Kaiming, H., Jian, S., & Xiaoou, T. (2011). Single Image Haze Removal Using Dark Channel Prior. *IEEE Trans Pattern Anal Mach Intell*, *33*(12), 2011, 2341–2353.

Kloss, G. K. (2009). Colour constancy using von kries transformations: colour constancy" goes to the lab". , 2009.

Kristan, M., Leonardis, A., Matas, J., Felsberg, M., Pflugfelder, R., Čehovin, L., Vojir, T., Häger, G., Lukežič, A., & Fernandez, G. (2016). The visual object tracking vot2016 challenge results. Springer.

Kristan, M., Matas, J., Leonardis, A., Felsberg, M., Cehovin, L., Fernandez, G., Vojir, T., Hager, G., Nebehay, G., & Pflugfelder, R. (2015). The visual object tracking vot2015 challenge results. *Proceedings of the IEEE International Conference on Computer Vision Workshops* (pp. 1–23), 2015.

Kuhl, F. P., & Giardina, C. R. (1982). Elliptic fourier features of a closed contour. *Computer graphics and image processing*, *18*(3), 1982, 236–258.

Kviatkovsky, I., Adam, A., & Rivlin, E. (2013). Color invariants for person reidentification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(7), 2013, 1622–1634.

Land, E. H., et al. (1977). *The retinex theory of color vision*. Citeseer.

Lee, H., Jeon, S., Yoon, I., & Paik, J. (2016). Recent advances in feature detectors and descriptors. *IEIE Transactions on Smart Processing & Computing*, *5*(3), 2016, 153–163.

Leutenegger, S., Chli, M., & Siegwart, R. Y. (2011). Brisk: Binary robust invariant scalable keypoints. *Computer Vision (ICCV), 2011 IEEE International Conference on* (pp. 2548–2555), IEEE, 2011.

Ling, H., & Jacobs, D. W. (2007). Shape classification using the inner-distance. *IEEE transactions on pattern analysis and machine intelligence*, *29*(2), 2007.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, Vol. 2 (pp. 1150–1157), Ieee, 1999.

Luan, X., Qi, W., Song, D., Chen, M., Zhu, T., & Wang, L. (2010). Illumination invariant color model for object recognition in robot soccer. *International Conference in Swarm Intelligence* (pp. 680–687), Springer, 2010.

M, A. R. S., Abhilash, S., & Supriya, M. H. (2016). A Comparative Study of Various Methods for Underwater Image Enhancement and Restoration. *6*(2), 2016, 30–33.

Maddern, W., Stewart, A., McManus, C., Upcroft, B., Churchill, W., & Newman, P. (2014). Illumination invariant imaging: Applications in robust vision-based localisation, mapping and classification for autonomous vehicles. *Proceedings of the Visual Place Recognition in Changing Environments Workshop, IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China*, Vol. 2 (p. 3), 2014.

Ojala, T., Pietikainen, M., & Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, *24*(7), 2002, 971–987.

Padmavathi, G., Subashini, P., Kumar, M. M., & Thakur, S. K. (2010). Comparison of filters used for underwater image pre-processing. *International Journal of Computer Science and Network Security*, *10*(1), 2010, 58–65.

Perona, P., & Malik, J. (1990). Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on pattern analysis and machine intelligence*, *12*(7), 1990, 629–639.

Pieropan, A., Björkman, M., Bergström, N., & Kragic, D. (2016). Feature descriptors for tracking by detection: a benchmark. *arXiv preprint arXiv:1607.06178*, , 2016.

Rasmussen, C. E. (2006). Gaussian processes for machine learning. , 2006.

Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). Orb: An efficient alternative to sift or surf. *Computer Vision (ICCV), 2011 IEEE International Conference on* (pp. 2564–2571), IEEE, 2011.

Sabhara, R. K., Lee, C.-P., & Lim, K.-M. (2013). Comparative study of hu moments and zernike moments in object recognition. *SmartCR*, *3*(3), 2013, 166–173.

Sande, K. E. A., et al. (2011). *Invariant color descriptors for efficient object recognition.*

Snoek, J. (2013). *Bayesian optimization and semiparametric models with applications to assistive technology.* PhD thesis, Citeseer.

Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems* (pp. 2951–2959), 2012.

Snoek, J., Swersky, K., Zemel, R. S., & Adams, R. P. (2014). Input warping for bayesian optimization of non-stationary functions. *ICML* (pp. 1674–1682), 2014.

Stalder, S., van Gool, L., & Avidan, S. (2012). *Paradigms in visual object tracking: Severin stalder.* ETH.

Suzuki, S., et al. (1985). Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, *30*(1), 1985, 32–46.

Swersky, K., Snoek, J., & Adams, R. P. (2013). Multi-task bayesian optimization. *Advances in neural information processing systems* (pp. 2004–2012), 2013.

Todt, E., & Torras, C. (2004). Detecting salient cues through illumination-invariant color ratios. *Robotics and Autonomous Systems*, *48*(2), 2004, 111–130.

Tola, E., Lepetit, V., & Fua, P. (2010). Daisy: An efficient dense descriptor applied to wide-baseline stereo. *IEEE transactions on pattern analysis and machine intelligence*, *32*(5), 2010, 815–830.

Uijlings, J. R., van de Sande, K. E., Gevers, T., & Smeulders, A. W. (2013). Selective search for object recognition. *International journal of computer vision*, *104*(2), 2013, 154–171.

Van De Sande, K., Gevers, T., & Snoek, C. (2010). Evaluating color descriptors for object and scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, *32*(9), 2010, 1582–1596.

Van De Weijer, J., & Gevers, T. (2005). Boosting saliency in color image features. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *1*(c), 2005, 365–372.

Van De Weijer, J., & Gevers, T. (2005). Color constancy based on the grey-edge hypothesis. *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, Vol. 2 (pp. II–722), IEEE, 2005.

Čehovin, L., Kristan, M., & Leonardis, A. (2014). Is my new tracker really better than yours? *WACV 2014: IEEE Winter Conference on Applications of Computer Vision*, Mar, 2014: IEEE.

Čehovin, L., Leonardis, A., & Kristan, M. (2016). Visual object tracking performance measures revisited.

von Kries, J. (1970). Influence of adaptation on the effects produced by luminous stimuli. *Sources of color vision*, , 1970, 109–119.

Walters, P., Sauder, N., Nezvadovitz, J., Voight, F., Gray, A., Schwartz, E. M., & Walters, P and Sauder, N and Nezvadovitz, J and Voight, F and Gray, A and Schwartz, E. (2014). SubjuGator 2014: Design and Implementation of a Modular, Fault Tolerant AUV. *AUVSI Foundation's 17th Annual RoboSub Competition, San Diego, CA*, , 2014, 1–8.

Winschel, A., Lienhart, R., & Eggert, C. (2016). Diversity in Object Proposals. (C), 2016.

Yang, K.-F., Gao, S.-B., & Li, Y.-J. (2015). Efficient illuminant estimation for color constancy using grey pixels. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2254–2263), 2015.

Zhang, J., Sclaroff, S., Lin, Z., Shen, X., Price, B., & Mech, R. (2015). Minimum barrier salient object detection at 80 fps. *Proceedings of the IEEE International Conference on Computer Vision* (pp. 1404–1412), 2015.

Zhu, W., Liang, S., Wei, Y., & Sun, J. (2014). Saliency optimization from robust background detection. *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2814–2821), 2014.

Zitnick, C. L., & Dollár, P. (2014). Edge boxes: Locating object proposals from edges. *European Conference on Computer Vision* (pp. 391–405), Springer, 2014.