UNIVERSITÀ
DEGLI STUDI
FIRENZE

DOTTORATO DI RICERCA IN
INFORMATICA SISTEMI E
TELECOMUNICAZIONI

CICLO XXVIII

COORDINATORE Prof. Luigi Chisci

Computer Vision Applied To Underwater Robotics

Settore Scientifico Disciplinare ING-INF/05

Autore:
Dott. Pazzaglia Fabio

Tutori:
Prof. Colombo Carlo

Prof. Allotta Benedetto

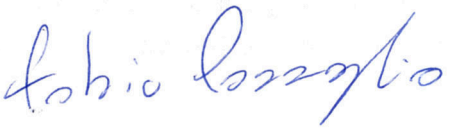
Coordinatore:
Prof. Chisci Luigi

Anni 2012/2015

Declaration of Authorship

I, Fabio PAZZAGLIA, declare that this thesis and the work presented in it are, to the best of my knowledge and belief, original and the result of my own investigations. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.



Abstract

Computer Vision Applied To Underwater Robotics

by Fabio Pazzaglia

Ocean and seafloors are today probably the less known and unexplored places on earth. Nowadays, the continuous technological improvements on underwater inspection offer new challenges and possibilities. Beside the classic acoustic sensors, modern cameras are playing an ever increasing role in autonomous underwater navigation. In particular, the capability to perform a context-driven navigation, based on what the vehicle is actually seeing on the seafloor, is of great interest in many research fields, spanning from marine archaeology and biology to environment preservation. Industrial companies on oil and gas or submarine cabling, also have a strong interest in underwater robotics. The peculiarities of the underwater environment offer new opportunities to computer vision and pattern analysis researchers.

This thesis analyses, discusses and extends computer vision techniques applied to the underwater environment. The main topic is the semantic classification of the seabed.

A framework that may actually be embedded in an underwater vehicle and made to work in real time during the navigation was developed. The first part of this work addresses the problem of semantic image labelling. For this purpose a deep analysis of feature sets and related classification algorithms was carried out.

The physical properties of light propagation in water need to be properly considered. Inspired by techniques for terrestrial single image dehazing, a new approach for underwater scenarios was developed. This approach is capable to significantly remove both the marine snow and the haze effects in images, and to effectively handle non-uniform and artificial lighting conditions.

By jointly combining the results of underwater classification and the physical modelling of light transmission in water, a new feature set, more robust and with better discriminative performance was defined. Experimental results confirmed the accuracy improvements, over the state-of-the-art obtained with the new feature set, in most critical environmental conditions.

This work is largely based on original images and data, acquired during the European project ARROWS. The novelties introduced by this thesis may represent a basis for future applications, stimulating novel directions for research in computer vision and its applications to the underwater environment.

Contents

List of Figures

List of Tables

To the person whose mind and smile fiLLed
my (rare) lacks of rationality.

Introduction

Today the use of *Autonomous Underwater Vehicles* (AUV) for environmental inspection is growing.
Modern technologies both in electronics and in mechanics allow the manufacturing and employment of these vehicles. On the other hand improvements in informatics extend their functionalities, hence the range of action that these vehicles are capable to perform.
More in detail this work is focused on analysing some possibilities offered by the optical sensors (cameras) that may be installed on it. In comparison to the acoustic sensors, the optical ones present today the higher innovation characteristics. Anyhow the information derived from underwater images is complementary to the one obtained by other devices as, for example, the acoustic ones.
Visual images can be used in the underwater environment for simple observation or qualitative analysis. Respect acoustic imaging, that still today represent the most employed technology, visual images offer several improvements about resolution, colour and definition.
The idea of achieving an automatic classification of the deeper seabed leads to two biggest advantages. The first one is the ability to assist researchers in the analysis of huge quantities of acquired videos and to allow the creation of a semantic database of the explored environment. In this way researchers may gain both in efficiency and in effectiveness in doing their job.
The second advantage related to the seabed automatic classification is the chance to perform a *context-driven* navigation. This requires algorithms that can be implemented on the AUV hardware and able to run in real-time during navigation with minimum delay respect of the acquisition frame rate.
The image classification method must be sufficiently generic and robust at the same time. In fact, based on the seabed automatic classification, the mission control module of the AUV may decide to investigate deeper certain interesting areas or rapidly move away when nothing of relevant is sensed.
Nowadays the related bibliography on underwater image classification is quite poor due

the fact that AUV technology is an expensive procedure and a research field in the middle between computer vision and robotics. The peculiarities of underwater environments make not easy to suite all the classic vision and pattern recognition algorithms already experimented in terrestrial scenarios. The principal reasons are hidden in the particular (natural) structures appearing in the seabed and in the light transmission properties.
Absorption and *scattering* are two effects directly linked to the light propagation in water and they need to be correctly handled. The atmospheric scattering is responsible of some types of degradations in acquired images that are easily observable when haze or fog are present.
Recently some techniques to deal with these phenomena in the terrestrial case have been presented. They do not require particular hardware but only a single image; their processing is based on theoretical models about the light propagation but until today their performance on underwater images are still low because they do not consider completely the involved physical effects.
In this thesis we address all the aforementioned themes. The chapter 1 is dedicated to provide some basis about underwater imaging and in particular the theoretical effects behind the light propagation in water medium.
Chapter 2 is dedicated to describe the application context of this work. In the first part the FP7-project *ARROWS*, that largely inspired this work, is presented within the state of the art of some image processing techniques applied to underwater robotics.
In the second part of the same chapter is instead introduced our developed framework for underwater seabed classification.
Chapter 3 deals with underwater haze removal techniques. Starting from classic methods applied to terrestrial images, the attention is then focused on the underwater world. We developed a new approach to *underwater dehazing* and several experiments are shown comparing other concurrent approaches. In the last part of this chapter other extensions and possible concrete applications of this technique are also proposed. In particular is shown how the haze is an actual cue directly related to the image depth.
Chapters 4 and 5 are completely dedicated to go in deep into the image classification task. The classification architecture is here described in detail. In the first of these two chapters the feature sets that we choose to compare are presented, justified and discussed. Parallel to this, all the datasets that we employed to conduct experiments are here deeply analysed. Considering the generalized lack of suitable data from other comparable works, these dataset were mostly created by ourselves from the ARROWS AUV images and videos. The chapter 5 describes in detail all the results achieved from our main experiments; in doing this, we considered several different parameters in our framework with the aim to get the best configuration.
The last part of this chapter is dedicated to provide a global summarizing discussion

regarding obtained results and comparing performances across many possibilities.
This leads us directly to the next succeeding chapter, in fact, by following the previous experience, the Chapter 6 presents a new feature descriptor to be used in our framework with textural images and explicitly realized for dealing with underwater images. A deep result analysis was conducted by comparing it with other classic feature sets showing the advantages and potential drawbacks of this new approach.
Finally, the last Chapter, the 7th, reports a summary and a discussion all across the themes that have been dealt in this work underlining future improvements and new field application.

Chapter 1

Underwater imaging: Description and basic tasks

This work starts with discussing problems and peculiarities of the underwater environment. It is a research field that closely involves both computer vision and photogrammetry. This latter field presents models that are very useful to deal with images that are often degraded due to the properties of the acquisition environment. As well described in [12], images can be processed by two different techniques leading one to an image restoration and the other one to an image enhancement. The last section of this chapter is dedicated to describe the bibliographic state of the art, including models to deal with these basic low-end image processing and to show obtained results.

1.1 Underwater environment

Computer vision for underwater imaging presents substantial differences with the terrestrial one. The basic geometry of optics can be found in [13]. Although the image formation model is the same, the different medium (water instead of air) involves physical effects which must be taken into account. Underwater images are generally characterized by a poor visibility caused by light which is exponentially attenuated. This fact limits the visibility—intended as the possibility to perceive electromagnetic waves in the visual spectrum from a given distance—that spans from 1 to about (in best conditions) 20 meters. Typical coefficients used to model the light attenuation strongly vary from bay, coastal and deep sea waters. The total pureness of water in nature is nothing more than an ideal concept.
In this work the focus is on underwater marine environment. With the exception of some

terrestrial water springs that have low interest for underwater robotics, deep waters are those that shows a higher purity.
There are slightly differences between sea or oceans in comparion of other environmental water as lakes or rivers. Even if some differences still remain among seas and oceans [14] [15]—for example caused by different environmental and life dynamics—their chemical composition and their interaction with light rays, present many affinities. Due to the properties of electromagnetic spectrum, the blue colour is characterized by more energy (shortest wavelength) than other colours. This makes underwater images mostly dominated by this colour that may change the appearance of the scene. Other times, is the water composition itself that strongly contributes to colour distortions. For example water, with a certain algae concentration, tends to cause a high presence of the green component.
Although deep waters are good for their clarity they also totally lack of natural illumination. Artificial light can be better controlled than the natural one and, by knowing its characteristics and its position, a camera could help to adjust contrast and recovering the true colour appearance. At the opposite, natural illumination covers the scene much more uniformly, avoiding brighter spots that may cause saturated spots with poorly illuminated surroundings.
It is possible to summarize the main problems in dealing with underwater images, as: *Range visibility*, *Illumination*, *Contrast attenuation*, *Colour changes* and *Noise*. They arise principally from:

- unknown variations in microscopic properties of the medium
- poor underwater lighting
- variations of absorption and scattering behaviour
- water transparency.

Standard computer vision techniques to underwater imaging may fail if we do not correctly deal before with these questions. A lot of specialized algorithms already exists (e.g. in [12]) and others will be shortly presented in this work. Some kind of distortion can be present both in the appearance and in the geometry of image.
Image enhancement uses qualitative subjective criteria to produce a more visually pleasant image and they do not rely on any physical model for the image formation. This kind of approaches are usually simpler and faster than restoring methods.

1.2 Physical background

The understanding of some basic concepts regarding the physics of underwater light propagation is crucial to deal with this kind of images. By definition an optically pure medium is one in which suspended particles are totally absent. This does not mean that there is total chemical pureness of an element—the compound can as well be a mixture—but that there is homogeneity in respect of the optical properties of the medium determined only by its atoms, molecules and ions without the presence of impurities. Most of the models about optics—included those specifically for underwater—often make assumptions concerning the hypothesis of pure medium with an isotropic behaviour.
The basic work about the propagation of electromagnetic waves in dielectric media was the Maxwell equations. Light passing from a medium to a different one is subject to the *reflection* and *refraction* phenomena. Considering a geometrical optical model, both effects determine a change in the direction of propagation of the electromagnetic wave and usually are properly referred depending on the size of its deviating angle.
Reflection is the part of electromagnetic wave that do not pass through the second medium; it is reflected and still continues its propagation in the medium of origin. Refraction, instead, is the part that can cross the interface and it is *transmitted* to the second medium. In physics and optical geometry the *Fresnel equations* are used to calculate these quantities [13].
In underwater the main focus is on refraction because it has a primary role on determining the *light attenuation* coefficient. Each material or compound is characterized by its own refraction index. For classical mediums, standardized value are used as reference even if they cannot fully handle the microscopic and local properties. In critical tasks, actual value needs to be experimentally determined in each case.
The main issues about light propagation—and more in general electromagnetic waves—in a dense medium like water, take the names of *absorption* and *scattering* effects. They were addressed for the first time in the twentieth century as testified in [16], [17]. In [18] and later in [19], the authors starts from discussing physical aspects to address the problem of simulation processes involved in underwater image formation. This latter work, in particular, is focused on the relationship between the image contrast and the received light power.
There is a trade-off that influences the perception and imaging in underwater. Basically, absorption phenomenon is mostly due to the actual water composition that cause a (selective) decreasing of energy of the light ray.
Scattering occurs when a light ray diverge from its straight-line path due to a variation in velocity caused by a medium change. Historically it is related to the theory developed after a series of studies and publications at the end of the nineteenth century by Lord

Rayleigh, and so today it is often referred also as *Rayleigh scattering*.
Actually this theory was developed for non-dense medium; scattering in liquids—although more evident than in gases—was discovered later and presents some differences with Rayleigh's model. This applies to particles that are small with respect to wavelengths of light, in particular the interaction within electromagnetic radiation and atoms or molecules of a impurity-free medium [17]. In water, scattering presents slightly different origins due to the presence of colloids or suspensions [20]. Note that this has not to be confused with the *diffraction* effect which only deals with the bending of waves around an obstacle or through an opening.
Absorption and scattering are generally synthesized as two separate coefficients, respectively a and b. Assuming an isotropic and homogeneous medium they are empirically grouped together and determine the so-called *attenuation coefficient*:

$$c = a + b \quad . \tag{1.1}$$

All the a, b and c coefficients express a constant decay which involves light intensity per unit of distance. Whit respect to the Beer-Lambert law the decay model of light intensity is related to the property of the water by an exponential relation. Let r be the distance from an object characterized by an irradiance of E_o, so the perceived $E(r)$ can be expressed as:

$$E(r) = E_o e^{-cr} \tag{1.2}$$

where the parameter c is the above-mentioned attenuation coefficient. This model can be further decomposed in a form that directly expresses the absorption and scattering coefficients:

$$E(r) = E_o e^{-ar} e^{-br}. \tag{1.3}$$

Although empirically determined, this is a useful model with high precision to describe the behaviour of thin and collimated light beam. Problems arise when, for reasons depending on the medium composition, the principal light beam lose its collimation and undesired rays may be scattered back with it. Figure 1.1 graphically shows this situation.

Different scattering events can be identified during the image formation model and they need to be considered in the previous equation 1.3. The scattering may be now expressed as the actual light beam—called the *direct beam*—with the superposition of other two contributions, the *forward-scattering* and the *back-scattering*. The difference between the forward and back-scatter concerns the angle whose the beam is deviated. In the first case there are small-angle deviations mostly caused by the intrinsic reflection

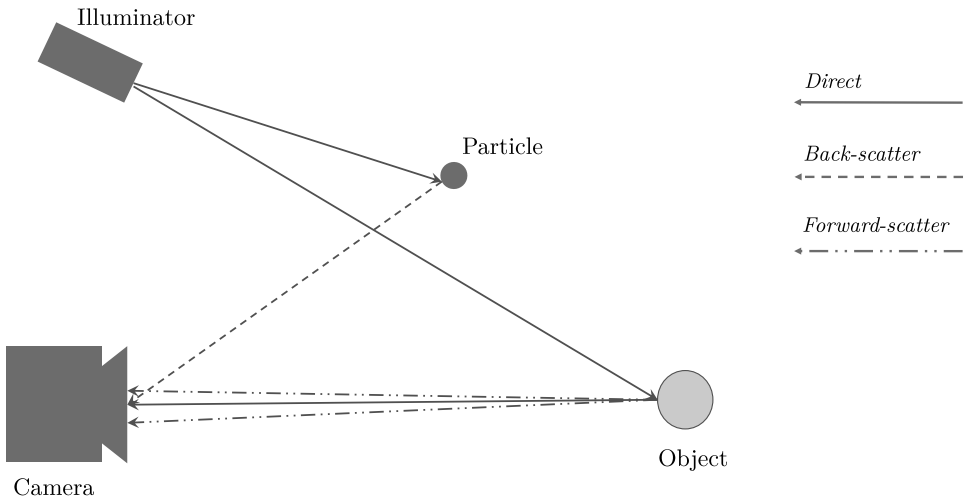


FIGURE 1.1: Direct light beam, forward and back-scattering effect.

properties of the towards an ideal, diffusely reflecting, Lambertian surface. The back-scatter instead is much more problematic to handle and ad it is caused by wide-angle reflections of light by other object different from the target.
In order to deal with this effects superposition another property $B(\theta)$ of the medium must be defined: the *volume scattering*. It is a function of the angle θ and must be integrated to obtain the total scattering b as reported in 1.4.

$$b = \int_o^\pi B(\theta) \sin\theta d\theta. \tag{1.4}$$

It theoretically considers all contributions coming from all directions.
Modelling the backward scatter is more complicated than the forward one because it requires the explicit volume scattering function. The four quantities a, b, c and $B(\theta)$ are those representing the properties of the medium. This resulting model can be used to predict the behaviour of light underwater in a high precise way.
The main practical effect is that the previous defined quantities vary with their location, and also are not constant over time.
As seen before, the light absorption is the other parameter that appears in the equation 1.3. This phenomenon involves the entire electromagnetic spectrum, and it is also due to the interactions, at very microscopic resolution, between radiation and atoms or molecules present into the medium. We are not interested here in deeply describing this relationships, but our focus is about the effects that they produce. In particular the application of the Beer-Lambert law depends on the existence of an analytical and well defined absorption coefficient a. The first remarkable thing is that the absorption

strictly depends on the wavelength of the radiation, usually indicated as $a(\lambda)$; so the behaviour is different across materials and among the components of a single light beam. As reported in [21] the absorption coefficient for water can be expressed as:

$$a(\lambda) = a_w(\lambda) + \sum a_x^*(\lambda)|x| \tag{1.5}$$

where a_w is the specific absorption of water itself. The sum over x involves other components that might be present in pure water in concentration $|x|$ and characterized by their coefficients a_x^*. In this way the absorption is a linear superposition of independent effects and determined by substances present in water. There are no general formulas to model these a values other than experimental measuring.

What is important to notice, both for scattering and absorption as well as other optical properties in underwater environment, is that their behaviour is always strictly related to the specific medium composition. This fact justifies the variability that we encounter in dealing with them.

The effects described above, scattering and absorption, are always wavelength dependent. The issues affecting the light propagation in the water are stronger than in the air.

Therefore, concerning the image formation process itself, the direct, back-scattered and forward scattered light constitute the three additive components of the total irradiance E_{tot}. So it can be mathematically expressed as:

$$E_{tot} = E_d + E_b + E_f. \tag{1.6}$$

This latter formula (1.6) is sufficient to understand phenomena and contributions that influence underwater imaging. As can be noticed this model assumes that the image formation process is affected by various degradations. In the ideal case, assuming a proper optics for the camera, only the irradiance E_d of the directed components hits the sensor, determining a clear, focused and noiseless image. The absorption effects may cause a change in colour and a poor contrast, but when the scattering effects become relevant, some more serious degradation can be encountered.

The presence of blur give rise to an image with serious lack in sharpness. This is not just a visual issue. Blurred image means that an information loss has happened, particularly at edges or high frequency components. The blur nature is very close to the one of a defocused image. Unlike this latter, that typically regards entire region of images and primary depends on the acquiring used optics, blur presents a much more localized and unpredictable behaviour. As shown in figure 1.2 the forward scattering is the principal responsible for blurred images. Due to small-angle deviations, light rays originated by near (non-adjacent) points can reach the image plane like a very poor collimated beam.

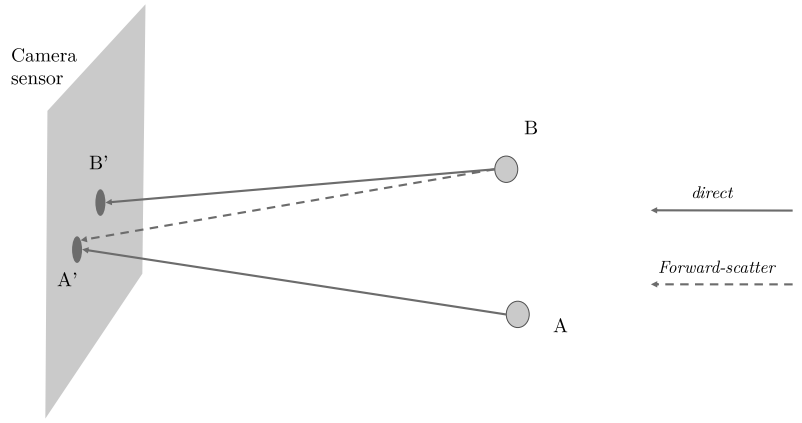


FIGURE 1.2: An effective representation of how the forward scattering affects the process of image formation.

So the radiance of one point will impress the corresponding sensor area together with a contribution that is a function of the radiance of other near forward-scattered points. The resulting image will be smooth and characterized by a loss in details.
The back-scatter instead is mostly responsible for the sparse noise on images, as depicted in figure 1.3. This kind of noise may have a *spike-form* in the easiest case. Using

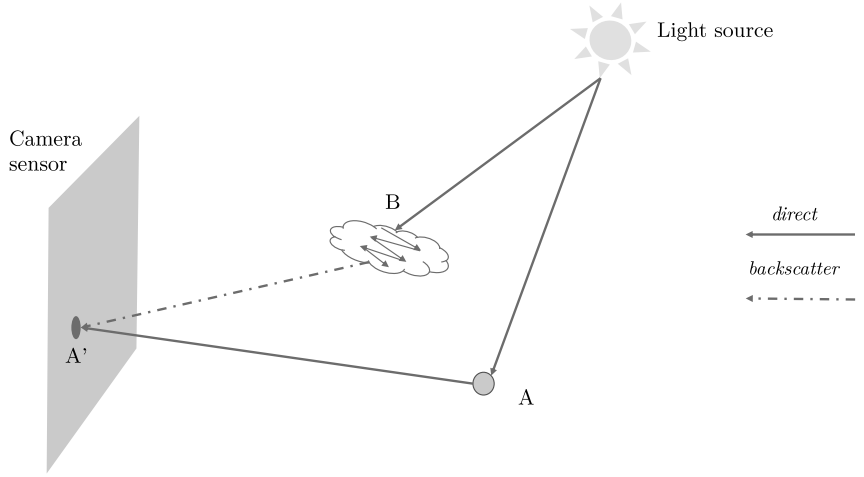


FIGURE 1.3: How the backward scattering cause noisy alteration in images.

classical image filtering methods it can be simply removed or attenuated in most cases. Nevertheless in other scenarios it can assume a much more diffuse form. When this

noise appears as an *haze-effect* it causes a huge information loss that is in general hard to recover. This effect will be deeply analysed in chapter 3 because it is an interesting research topic and needs a special model. In figure 1.4 both effects derived from forward- and backward-scattering are present. In particular, on the middle-top of this image a noisy haze—mostly due to interaction with sunlight—makes difficult to recover the real appearance. On the bottom of image, instead, a discrete amount of blur makes the image quite smooth.

For further details in [19] the crucial quantities E_d,E_b and E_f as well the analytical

FIGURE 1.4: An image that shows the effect of common noise due to underwater environment.

formulas that link them are discussed in deep with the definition of an expression for each component of total irradiance.

1.3 Geometrical aspects and effects

There are several studies about the geometrical distortion induced by the water medium [22], [23]. The refraction effects—the bending of a wave when it enters in a different medium—is the main cause. In the classical photography, light rays are conveyed to the image CCD or CMOS sensor, through a lens. Considering well-designed and low-distortion lens they can well approximate the *pinhole camera*, the underlying model today in most of the visual geometry. This model is valid until light rays pass through one single medium, more or less homogeneous, during the trip from the emitting source and the camera. Typically this medium is air, but for underwater imaging it is obviously the water. As seen in the previous section about underwater physics, the water-medium has a higher refraction index and its presence in nature is characterized by stronger differences

in chemical composition that can affect the physical interaction with electromagnetic waves. The effective problems arise from the fact that to take underwater images—especially at non-trivial depth—we need a waterproof housing for cameras. These camera containers, commonly realized with a thick layer of a transparent material, determine a three-medium system passed through by rays, as reported in figure 1.5.

The optical system is composed by two parts that affect paths of optical rays between

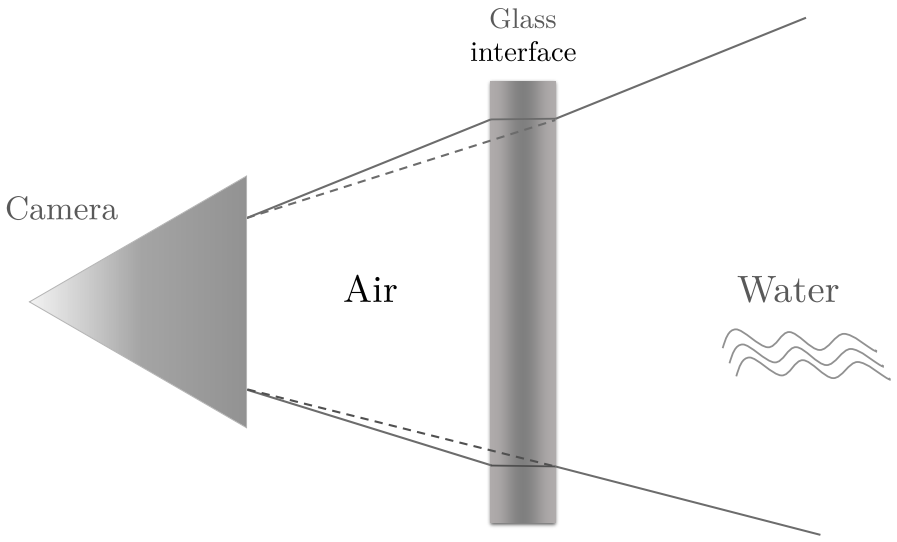


FIGURE 1.5: Situation in which light rays are passing through a three medium system before reaching the camera sensor.

objects and their images. The cover lenses are in the front of the camera and are assembled on the cover of the housing that isolate the system against high pressure in a deep-water environment. The shape of the cover lenses could be of two types: convex or plane-concave. In general the use of concave lenses can increase the field of view and reduce some kind of distortion induced by water.

There are two different interfaces, *water/glass* and *glass/air*, that deviate the light rays from their original direction causing two different angle of refraction before reaching the camera. From the geometrical point of view the camera needs some specific calibration procedure or correction as reported in [24] and [25] to take this effect into account. A crucial step if we have to employ underwater algorithms for *structure from motion*, *visual odometry*, *3D stereo reconstruction*. Despite algorithms and models, to best overcome these issues is fundamental using a specific-designed housing/lens system for underwater images. The *water-glass-air* distortion may vary with water depth and composition, so experiments and compromises are needed in order to best accomplish specific tasks.

Even if we are not directly interested in recovering geometrical features from the image,

if it is not treated properly may affect also the direct appearance of captured images, spanning from a slightly blur noise to a more strong defocus. In both cases the image may lose definition and more important may lose information that hardly will be otherwise re-established.

1.4 Image restoration

The final aim of image restoration is to recover a degraded image by considering the image formation process and by using a specific model of the occurred degradations.
This kind of methods can improve the image better than other enhancements under the hypothesis of choosing the right model and overall they require to correctly set some parameters. The knowledge of how to tune these values is not trivial and commonly treated as an experimental matter. The physics about the image formation can help to use a correct model for the environment (as reported in Section 1.2) and some knowledge about the noise statistics are needed in order to approximate certain effects. Without entering into the particular noise distribution models, what is important to note is the difference between *additive* or *multiplicative* noise. In image recovering the multiplicative factor is usually regarded as a more general *degradation function*, which better incorporates the effects of the imaging system and the medium. So, the image $i(x, y)$ can be regarded as the function:

$$i(x, y) = j(x, y) * h(x, y, \theta_1^h, \ldots, \theta_n^h) + n(x, y, \theta_1^n, \ldots, \theta_n^n) \tag{1.7}$$

where $j(x, y)$ is the real value at point (x, y), $h(\cdot)$ and $n(\cdot)$ are respectively the degradation and the noise function, both with a set of parameters θ^h and θ^n. The $h(\cdot)$ is a *point spread function* (PSF) and describes the response of an imaging system to a point source of an object (in [26] is presented a brief discussion and comparison about different point spread functions). The fist two terms are convolved together so passing to the frequency-domain this become a simple multiplication.

$$I(u, v) = J(u, v)H(u, v, \cdot) + N(u, v, \cdot) \tag{1.8}$$

Many restoration methods are created to consider this latter domain.
Because the strong degradation usually induced some models for underwater imaging was realized.
Better is the knowledge that we have about the degradation function, better are the restoration results. In practical cases, there is insufficient knowledge about the degradation and it must be estimated and modelled. Sources of degradation in underwater

imaging includes turbidity, floating particles and the aforementioned optical properties of light propagation in water.
In [27] the authors presented a self-tuning restoration filter based on a simplified version of the Jaffe-McGlamery image formation model. Two assumptions are made in order to design the restoration filter. The first one is a uniform illumination and the second one is to consider into the model only the forward component (i.e. ignoring back scattering). This is reasonable until the concentration of particulate matter generating limited, even if appreciable, effects. So, a low backscattering component and a condition of shallow-water represent the optimal environment for applying this technique.
They assessed quantitatively the benefits of the self-tuning filter as a preprocessing step for a subsequent classification, to check where an image contains or not man-made objects.
In [28], authors deal with the polarization effect to compensate for visibility degradation. The proposed algorithm is based on a couple of images taken through a polarizer filter with different orientations. Even if the raw images are characterized by limited contrast, the light differences can lead anyhow to visibility improvements. In [29] a same approach was further improved to both recover underwater visibility and provide a coarse 3D estimation about the 3D geometry. Also this method requires specific hardware and it is based on multiple images of the scene.
As will be shown the method developed in this work (see Chapter 3) uses, instead, a different approach, derived from the so-called *dehazing* techniques and employing only a single image of the scene.

1.5 Image enhancements and performance measures

Differently from the image restoration techniques, the enhancement methods take into account the image formation process, without any relevant a priori knowledge. This means more general approaches substantially independent from the image scenario.
Image enhancement is basically a non-specific process used to improve the visual quality or perceptual information in images mostly directed to human viewers.
In underwater images it has to be noticed that as depth increases, depending on their wavelength some colour components tend to disappear. Although image enhancement theoretically abstract from the model, the considered environmental properties cannot be totally forgotten and in the scientific bibliography there are specific works about underwater image enhancement techniques. In [1] is proposed a multi-step filtering approach specifically developed to be employed in underwater vehicles as first image treatment to reduce several common degradation effects. The pipeline is composed by

nine steps mainly focused on dealing with non-uniform illumination, noise suppression and colour adjustments. Figure 1.6 shows some results obtained with this approach. A similar approach based on multiple filtering and contrast equalization is proposed in

FIGURE 1.6: Results achieved by the method proposed by Bazeille et al. in [1] (image from the original work).

[30]. Here the scattering and absorption phenomena are explicitly mentioned but they are not directly modelled as in the restoration techniques.

Another technique for colour enhancement was presented in [31] and differently from the aforementioned is based on perceptual approach and inspired by lightness and colour constancy properties. This method is also suggested as preliminary step to improve segmentation.

In [32] the problem of colour enhancement and restoration is addressed as an energy minimization problem, modelling the image as a Markov Random Fields but employing more than one image to actually work.

Another theoretical method to enhance underwater image in the scene with planar surfaces and in presence of non-uniform illumination and low contrast is those proposed in [33].

More recently in [34], [35] and [36] other novel enhancement approaches have been presented and mostly aimed to actually guarantee image details preservation.

A single image strategy is proposed in [37]; here, without a specific hardware or knowledge about the environment the authors performs a white balancing and a noise reduction (that also can work with video and maintaining temporal coherence) to actually enhance the image appearance. In [38], the authors propose an algorithm for the enhancement (and restoration) of blurred underwater images substantially based on median filter.

Finally, in [39], instead, a comparative analysis of three different enhancements techniques—contrast stretching, histogram equalization and contrast limited adaptive histogram equalization—was carried out. Without considering a particular colour space, the adaptive equalization is the approach that seems provide better results.

During time many different methods for image quality evaluation have been proposed and analysed. As well as for the restoration techniques, image enhancements are difficult to measure in comparison of natural images for at least two reason.

The first is the difficulty with determining a commonly adopted and objective method for the evaluation of perceived quality on a given image. A "well-enhanced" (or restored) image is hard to establish univocally. Anyhow, even considering less subjective measures related to image information and/or noise the second issue is instead related to have available a reference image to actually calculate these values.

Classically, *Peak Signal to Noise Ratio* and *Mean Squared Error* are the most widely used measures when a reference image is available.

Even if there is not a prevailing one, in bibliography various attempts to define new specific indices for quality assessment were introduced. Clearly it is not a problem exclusively regarding the underwater environment. In [40], it is proposed a methodology based on simulations and considering the well known Jaffe-McGlamerys model for underwater images. In particular authors explicitly mention the problem related to image noise, marine suspended particles, light attenuation and non-uniform illumination. Actually this is an a priori evaluating strategy that enables to benchmark algorithms suitability for underwater conditions. In other words it may be employed for a rigorous pre-evaluation and comparison algorithms for underwater applications and only on synthetic data.

In [41], it was presented an approach that uses edges and image sharpness to evaluate the image quality after its processing. In particular an image quality metric is defined and specifically tuned to better respond to the environmental parameters. Finally in [42] is reported a short review about the traditional approaches based on error-sensitivity to image quality assessment. Starting from the intrinsic limitation of these approaches, the authors proposed a new different measure, capable to handle with structural similarities and called *Structural Similarity Index*.

In conclusion it must be pointed out that unless a valid reference image there are no objective and fully reliable ways to asses image quality, especially in underwater images. For general applications a straight direct comparison still remain the preferred way in most of the recent works.

Chapter 2

Underwater image processing framework

2.1 Motivation and State of the art

Nowadays the use of *Autonomous Underwater Vehicles* (AUVs) for environmental submarine inspection is growing. Modern technologies on both mechanic and electronic fields allow this kind of vehicles to be ever more used for environmental underwater inspections.
The availability of powerful batteries and low absorption components makes easier the project and the actual hardware realization. On the other side, smaller size computer units and their growing performance enable the chance to guide the AUV in accomplish more sophisticated tasks.
At their essence, AUVs are a submarine vehicle that is capable to navigate and maybe to take some actions in a total autonomous way. They differ from most common *Remotely Operated Vehicles* (ROV)—sometimes referred also as ROUV, Remotely Operated Underwater Vehicles—for basically the fact that they have not a direct physical connection with a base station, as for example a supporting boat, that provides both power and instructions. ROVs are typically guided from remote places or anyhow supervised by humans on the basis of information gathered by sensors installed on them.
In underwater medium the communication channels are limited; radio or electromagnetic signal transmissions are characterized by a short range so that the ROV are actually linked to the base station with a cable that clearly limits the movements capabilities of the vehicle.
On the other hand AUVs have not a direct link with a base station and they need to

take internally their decisions about the actions that have to undertaken. Today, the operative possibilities offered by ROVs are still greater than AUVs—because the fact that they are largely adopted in marine oil platforms—which are instead employed mostly in environmental inspections ([43] and [44]).
An AUV is typically provided of several different sensors, some used for navigation and others for inspection. In this work we deal with images and our focus is on optical sensors. Taking underwater images requires that the vehicle is able to navigate sufficiently close (few meters, depending on actual environment) the sea bottom because the visual signal quickly decreases with the distance.
During underwater inspections, AUV enables researchers to acquire potentially a lot of visual data. In the simplest case the vehicle can be programmed in advance to follow some navigation plans and to acquire images during all its journey. These videos may be after seen and analysed by humans with or without the assistance of (semi)automatic software. The drawbacks of such approaches are evident due to the limited possibilities for actually drive the inspection.
To directly provide the AUVs with the capabilities of real time image analysis is a challenging task that may improve a better context driven navigation and also make the inspection more effective and hence all the entire process will be more efficient.
As said before we are focused on image analysis but more in general the topic of autonomous context driven navigation also takes profit by other sensors.
In this work particular attention is given to inspect the possibility to actual classify images directly "on-board" and in real time during the navigation. As major advantage this may lead to perform targeted searches that might be suited in several ways by researchers.
The main application fields are marine biology, underwater archaeology and environmental preservation.
As will be explained in a following section the aim of this *classification* is primary directed to the overall environment and it is not—in its original conception—properly directed to particular objects.
Even if our developed processing framework can be used also for the post processing phase, working in real time during the navigation determines a low computational time cost as the main requirement.
The navigation is a key challenge that needs to be improved ([45]). The output of the classification task is proactive in the sense that it may change the navigation plan because the main software module that controls the AUV navigation uses these information to take decisions.
With the aim to gather data from underwater environments, the classification allows to automatically label acquired images not only in relation to the location but also to the

content. This may save a lot of time in post mission activities undertaken by researchers. Anyhow, our framework architecture and its output will be discussed more in depth in following sections.

Looking at the related bibliography, that sometimes has to be found in the middle between robotic and vision, there are some relevant works that may be taken into consideration.
A good and concise review of optical imaging for underwater vehicles can be found in [46]. In [47] there is instead a review of more recent underwater system technologies with a close look to the vehicle navigation characteristics.
One of the earliest work is reported in [48], where a vision system is presented for predictive segmentation especially designed for underwater robot tasks.
To testify how the problem of identify some common underwater patterns, in [49] is presented a vision system for automatic detection of ripple pattern starting from ROV acquired videos. The idea is to skip those images that are uninformative—as for example those with only sand-related patterns—about the actual interests of the mission.
Another similar approach, this time used for automatic change detection is dealt in [50]. The same authors in [51] extend the previous approach to work in real time applications.
Walther et al. report in [52] an automated system for (post-) processing underwater videos that is able to detect and track objects (fish actually) that might be of potential interest. In [53] another method for identification of underwater objects was proposed, this time with a more general approach and mostly based on the analysis of perceived colours of the scene.
Actually not all these referenced works are explicitly aimed to support the AUV mission. Most of them are not suitable to be used during navigation, both for their computational demands and temporal costs.
More recently in [54] the problem of underwater habitats classification, starting from video. In [55], instead, was presented a series of experiments for automated species detection of algae from AUV acquired data. Here the authors adopted a scheme that is, in comparison of the previous ones, closer to our adopted classification framework and that will be described in the next section.

2.2 The ARROWS project

The *ARROWS* (ARchaeological RObot systems for the World's Seas, Fig. 2.1) project ([56]) is a collaborative project coordinated by the *University of Florence* (Italy) with partners from academia and industries from all across the Europe.

It started on 2012 by following also the experience of another underwater project,

FIGURE 2.1: The ARROWS logo.

named THESAURUS (TecnicHe per l'Esplorazione Sottomarina Archeologica mediante l'Utilizzo di Robot aUtonomi in Sciami) project ([57]) characterized by similar aims and in which the University of Florence has also been involved.
The project ARROWS was partially funded by the European Commission during the *7th Framework Programme* (FP7), and it has been ended in middle 2015.
From the big picture perspective, the Arrows proposal was to develop low-cost autonomous underwater vehicle (AUV) technologies directed to support and improve archaeological underwater operations with cost-saving solutions and the use of new (modern) technologies.
Although this project was explicitly related to the archaeological field, most of the developed solutions are clearly suitable also for other kind of applications as for example, environmental inspections and seabed mapping. Anyhow, most of the ARROWS tasks were explicitly referenced to the cultural underwater heritage scenario with the presence of a number of archaeologists into the partnership.
Beside the vehicle development, overall some of the main actions enabled by the activities of this project regards:

- Horizontal large-area surveys with the acquisition of data originated from a number of different vehicle sensors.
- High quality seabed maps both for reconstruction and geolocalization activities.
- Small low cost vehicles for penetration in hard and small environments.
- A fully and semi-automated data analysis tool for enabling the possibility of a context driven navigation and to realize a semantic labelled map of the explored environment.
- 3D reconstruction and mixed reality environments for virtual explorations.

The ARROWS project developed and evaluated low-cost AUVs capable of achieve systematic surveys of the seabed by programmable missions both close the sea surface and at some hundred meters depth. AUVs are realized to be modular (i.e. with variable payload that can be embedded on them) and able to work in a collaborative way with

other similar vehicles.
The acquisition hardware/software installed on the vehicle was specifically developed for this project;
it is schematically composed by two cameras with proprietary interface and driver. All the vision modules (hence not only the attentive vision) run as a *ROS* (Robot Operating System) node and are capable to read activation and deactivation signals originated from higher level modules, that drive the hardware behaviour. Other information, like the status, are instead implemented with using the standard ROS routines.
The acquisition module is able to work both in mono and stereo configuration depending on the specified task. In all cases a video is recorded meant to be used in post-processing applications.
To store and efficiently process every video frame a *producer-consumer* pattern was used so that each recorded frame can be asynchronously accessed, locked and shared by processes that might need the resource. Because the buffer for the temporary frame storage is limited and each location serially accessed, there is the necessity that all visual tasks may perform in sufficiently short time related to the desired frame rate processing.
The acquired image resolution may vary from 2040x2040 to 1020x512 depending on particular application.
In the context of the present work the qualities and technological improvements carried out by vehicles themselves are not reviewed. More detailed information can be found in ARROWS related bibliography, as for example [58] and [59].
The activities linked to the semantic labelling and seabed classification, are those that are here analyzed in depth.
During its mission the AUV examines the seabed, looking for areas of potential interest. With limited or none connection to a base station, the vehicle must be capable to perform analysis and to take decisions by itself. This leads to a context-driven navigation based on observing the local seabed appearance. In particular, considering the wide extension of seas and the relative sparse and limited areas of interest, the vehicle should have the capability to rapidly move away when for example uniform sandy seabed is present. On the other hand, when an area of potential interest is reached, the AUV should have the ability to reduce its navigation speed and actively perform a deeper investigation that might also need to activate some other more specific sensors.
From the point of view taken by the mission control module, the optical sensors are a choice between many others dedicated to the navigation. In particular acoustic sensors are more capable to make wide-range analysis of the seabed while the optical one can be used to go deeply in potentially target areas. Although their limited range, optical informations are in general higher both in number and in quality instead of those gathered by different sensors.

Even if the vehicle is equipped with two cameras, actually to comply with timing constraints imposed by navigation, only the output of one of them is used. In fact the two cameras are installed to be used in other vision task that requires a stereo configuration as for example the visual SLAM and 3D reconstructions.
A challenging aim was also to use all the gathered visual information to realize a 3D virtual submerged underwater world, built using all the data acquired and which can allow general users to interactively explore the environment.
After each mission the (archaeological) researchers are in this way provided with annotated maps of the seabed together with the complete acquired underwater videos. These labels may quickly indicate the environmental composition or interesting areas.
To perform the seabed classification and consequently enable the possibility of an attentive navigation system, the seabed is analysed with a texture-oriented algorithm. After a supervised training stage—performed offline during the navigation— the system learns, through examples, a dictionary of textures and contextually acquires the ability to find similarities and to discriminate across learned environments by inferring each time the most likely. This is actually the real link between the perceived images and its virtual ontology based representation (Figure 2.2).

The information gathered by the seabed classification is thought to feed a *distributed*

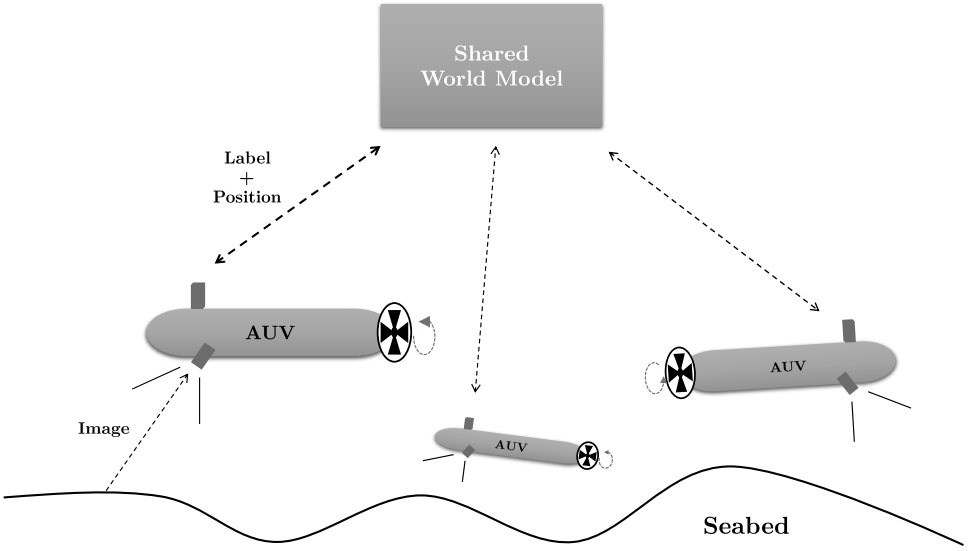


FIGURE 2.2: The vehicles and the shared world-model representation realized by geolocalized labels of classified places.

world model. More in deep, through a specific defined ontology the vehicle is able to understand and remember the world already inspected. Starting from an almost empty model, this world representation is enriched by information carried out by the analysis

of several sensor outputs and the geolocalization dedicated hardware. The use of this world model can be shared across different entities and continuously augmented during different mission by also employing different vehicles. By using supervised machine learning technologies, the seabed classification results are effectively related also to the pertinence of the used dataset to perform the training phase.
Furthermore this map may be used also to favour other aspects of the post-processing activities as for example:

- to favour the geolocalization in a virtual navigation environment
- to allow semantic (automatic) searches inside the acquired videos (e.g. Which are the frames characterized by the presence of sand or marine vegetation?)
- to spatially localize each video frame.

Overall the key step of this mapping task is also to handle and organize in an automatic manner the huge amount of information collected by every AUV mission and to enable an easy and effective retrieval.
Other than a description of seabed, in the post-processing phase might be also achieved a more specific object detection task, employing usual object classification and retrieval algorithms (e.g. [60], [61] and [62]).
Anyhow it must be noticed that in underwater environments, poorly affected by human interventions, the notion of "object" may be unclear and sometimes even trivial.

2.3 Task definition

The attentive vision module of the AUV uses the visual data provided by the optical sensors to identify potential interesting area during the mission. The optical sensors are complementary to other acoustic modules that may be found on the vehicle, as for example the side-scan sonar and the output can be combined together in order to gather a wider analysis. Clearly, acoustic sensors are focused on the inspection from longer distance while optical images are obtained by a more close range acquisitions and may capture additional data from the scene. In this environment the most important are the informations about image textures.
These data can be analysed and processed with machine learning techniques to classify the seabed and to help in identifying possible areas of interest.
In developing the software module responsible of such processing the limitations imposed by the hardware vehicle specifications have to be taken into account together with the

real-time execution constraints. In particular the time spent for processing each image has to be almost constant or however there must not be an accumulation of processing delay during mission. The attentive vision module can be synthesized as in figure 2.3. The results of these computations are employed by higher software modules to take

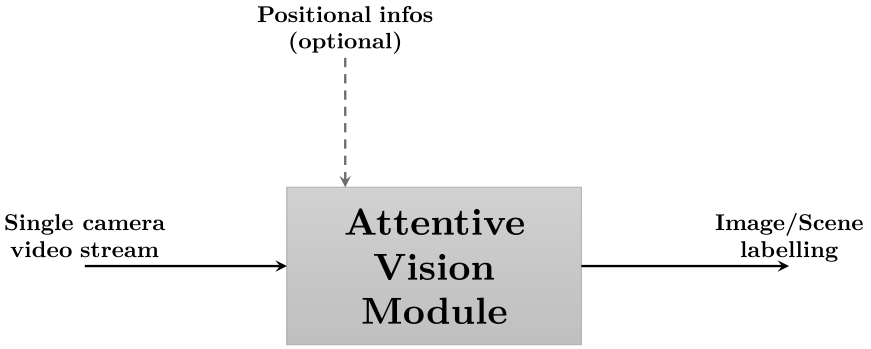


FIGURE 2.3: Concise representation of the attentive vision software module.

decisions and they contribute to create a semantic representation of the underwater environment. This latter point may be significantly achieved by combining the environmental labels with the geo-localized information based on robust position estimates (e.g. SLAM techniques).

The classification of known underwater environments for the attentive navigation requires to classify images according to predefined common underwater classes. In particular, it has been implemented a supervised learning approach based on *Support Vector Machines* classification, which is one of the most general state-of-the-art approach.

To have an effective classification of the seabed it is crucial the use of a training set which can be selected according to the task and environmental aspects. In fact these latter ones may vary from place to place, and there is no easy way to gather all them together. The classical machine learning framework for supervised classification is depicted schematically in Figure 2.4; it mainly consists of two steps, the training phase and the actual classification.

In the training phase the algorithm learns the different classes according to a provided dataset, consisting of labelled examples of image patches representing the desired classification. A good selection of datasets is crucial for the subsequent phase.

While the computational time needed by the training phase might be high it has not a high impact on the execution during navigation because it is a one-time job achieved before the actual mission.

In the classification step instead, time constraints must be taken into account during

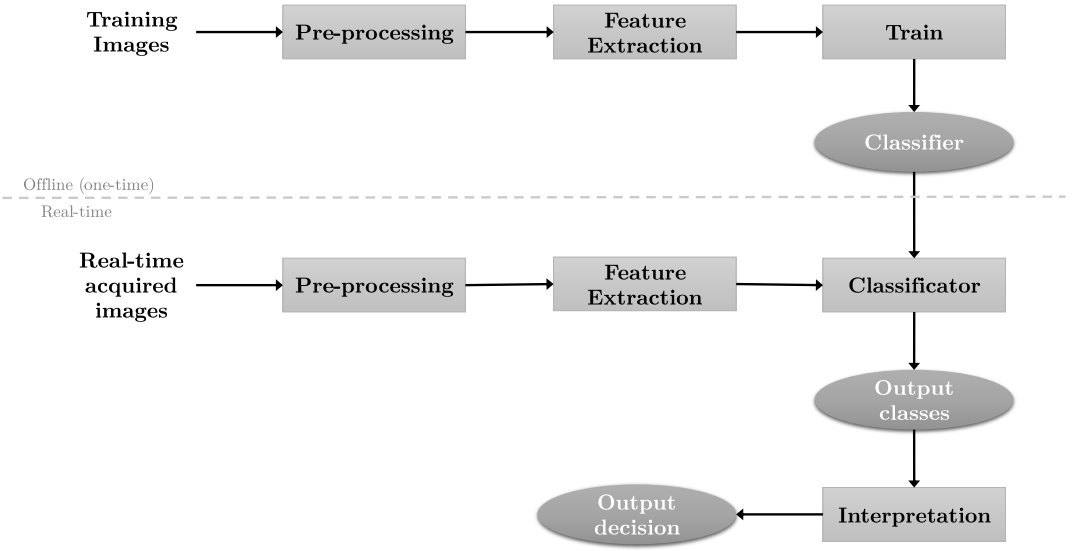


FIGURE 2.4: Classic scheme of a supervised learning approach.

the AUV mission.

As can be noted in Figure 2.4, both the training and the classification steps share common image pre-processing and feature extraction execution blocks.

The choice of a set of suitable, efficient and discriminative features is crucial for every machine learning algorithm and actually it might be the most time consuming step both in training and in classification step. In details, the adopted approach works by segment firstly the image into a number of regions. These region may be characterized by variable or fixed size depending the choice to (pre)process the image with a segmentation algorithm or just to take equally sized windows. We notice that, dependently on the implementation and also the available data, images might be previously scaled, filtered or depending on the available computational time, a pyramidal representation of the image might be used.

In any case, each region is then processed to extract its features successively processed by SVM.

2.3.1 Classification Architecture

To perform the activities of seabed classification an *ad hoc* software framework was developed for the attentive vision module.

As previously said the vehicle needs to perform the classification of seabed in real time during navigation and with time constraints linked to the entire AUV hw/sw architecture and taking care of other vision modules. This will be referred as the *online* classification

task.

Otherwise based on the acquired videos during navigation a post-mission classification can be also conducted. This will be instead refereed as the *offline* classification task in the developed framework. The usefulness of this approach is the chance to conduct more accurate classification without time constraints also in using higher number of classes, more sophisticated features or simply because the needs to re-examine a video (or a portion of it) for conducting a different analysis by following other criteria.

Using both the online and the offline classification the training step is almost the same and follows the block diagram reported in Figure 2.5.

The input of this learning algorithm is a number of labelled examples representing

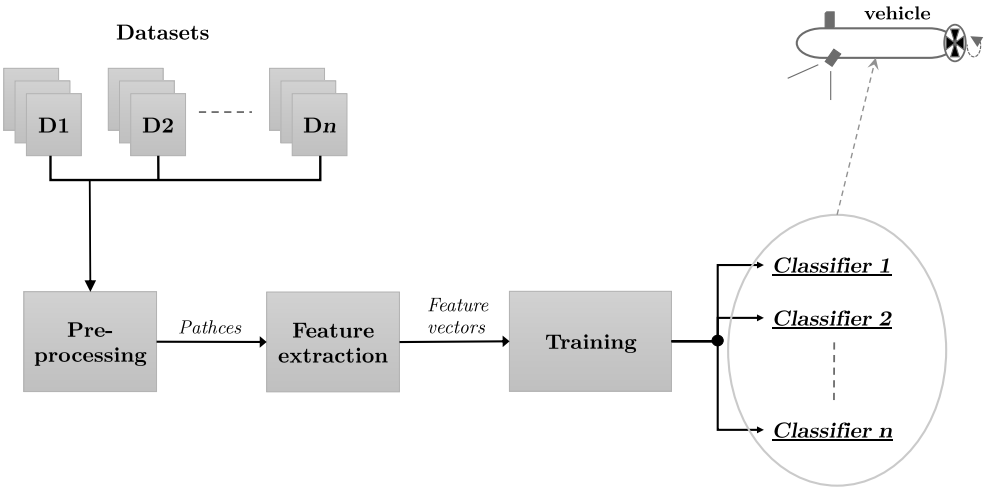


FIGURE 2.5: Block diagram representation of software used to train the classifier.

the diverse classes that have to be classified (for more details about the classification algorithm see Section 4.4.2). Each group of image represents a dataset with types and number of classes not necessarily coincident.

Normally every dataset is used to train a single classifiers. Anyhow multiple datasets can be also grouped together.

The reason behind the choice of training several independent classifiers is due to the highly variable characteristics of underwater environments. With the support of practical experiments we noted that realizing a single classifier leads in general to lower performance due not only to differences in context but also on intra-class variations.

The employment of task-specific datasets needs a choice that has to be done, autonomously by researchers or automatically by the aid of an algorithm based on the scene analysis.

Referencing again the figure 2.5 and following the input image line, there is the *preprocessing* block. Here are grouped all the tasks designed for normalizing the image with both colour balancing and filtering approaches. Note that the dataset examples are here composed by patches and not by entire images.
The next block is the feature extraction, that is responsible of transforming the input image in a feature vector with a (predefined) number of component depending the particular feature set actually used. The employed feature set might be changed but anyhow it must ensure coherence with the corresponding classification task.
To this regard we may notice that all our software architecture is completely modular and the operation inside each block might be changed with maintaining the same block interfaces.
The feature vectors are the inputs to the actual training phase based on *Support Vector Machines*. The output of the SVM training is a classifier that may be then used to actually classify the image both directly on-board (online) and on previously acquired videos (offline). In general the training phase is much more slower than the time required by the successive classification step. It mostly depends on the type and number of extracted feature and the number of examples given as input.
Switching to the classification step, we already said that our framework contemplates two execution possibilities, online and offline. Both are described in the next section.

2.3.2 Online vs offline approach

Figure 2.6 shows the internal software diagram for the attentive online vision module. The architecture is complementary to the training one.
The AUV camera acquired images that (excluding the initial shared buffer management) are directly passed through the preprocessing task.
Other than the (optional) image normalization task, here the image is preliminary segmented in a number of patches. This segmentation, depending on the adopted strategy, must be carried out on two ways: *fixed window* or *variable window*.
In the first case the image is divided into n rectangular windows of predefined size and regardless their content.
With the second choice, instead, the image is segmented in windows with a size dependent on some uniformity property measures of the considered area. The proper advantages of using fixed versus variable window size are discussed more in detail in Chapter 4.
Anyhow despite consequences and final performance achieved this second step, carried out by a fast *QuadTree* segmentation is more time consuming and hence might reduce

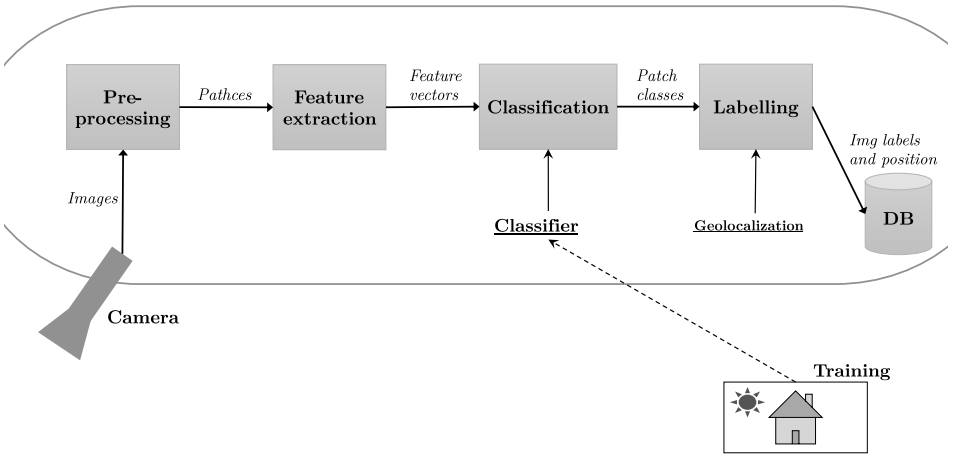


FIGURE 2.6: The online attentive vision module; block diagram representation.

the frame rate of processed images. In fact, as for the training phase, the time spent is primarily related to the feature extraction and the correspondent vector dimension. In this case also the time required to perform the initial segmentation plays an important role that is further dependent on the number of extracted patches; clearly the adoption of a fixed size window strategy guarantees a more stable execution.

It can be noticed that the minimum frame rate necessary to actually conduct a fine seabed classification depends also in vehicle-related characteristics, as its speed and the camera distance from the seabed.

By continuing to follow the block diagram classification scheme, the set of patches is then passed to the feature extraction task. This block is exactely the same as used during the training phase and depending on the chosen features it generates the feature vectors that are successively classified in the successive task.

Other than the feature vectors of the current images, the classification block use the classifier trained in the learning phase.

Even if the time consumed to perform the classification task is limited (for example in relation to the time used by previous tasks in the diagram) this online classification employ only a single classifier each time, that has to be selected before the mission starts. One label is then assigned to every classified patch and correspondent to one of the known classes. Regarding all the different labels that might be associated to each image, only the most relevant (in terms of occurrences) are selected and the image is classified with those labels conveniently weighted.

Together with the semantic class label and before to store these information, again this last task associates to every image the actual AUV geolocalized position. To perform

more accurate analysis the developed framework can be employed in the *offline* manner. The synthesised block diagram is reported in figure 2.7. In this configuration, that

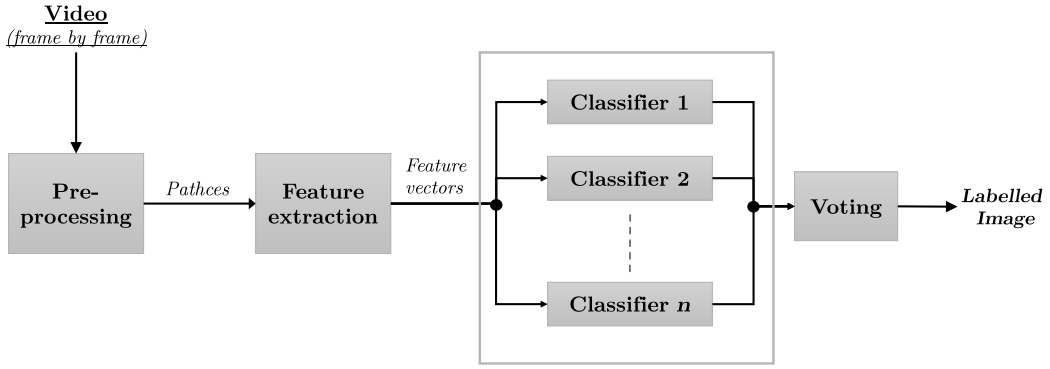


FIGURE 2.7: The offline attentive vision module; block diagram representation of the *offline* module.

works when the mission is ended, the input is properly a video sequence registered by the vehicle. In practice there are no many differences in relation with the case of real time acquisition and in fact the first blocks are mostly the same that are present in the online version.

Clearly, without particular timing constraints, theoretically more complex feature sets or preprocessing filtering operation can be used. The actual difference is in the proper classification block.

To achieve an accurate discrimination, multiple classifiers may be used in parallel to process every image patch. The *multiclassifier* approach may improve the performance of a single classifier (e.g. [63]). The effectiveness of a classifier combination is clearly related to the achieved independence between the single ones. Then through a voting scheme or fusion of confidence score (if available) the labels with higher rating can be chosen. In particular the approach of training multiple classifiers over different datasets is useful in cases where there are poor information about the investigated environment or rapid changes might be expected in the seabed appearance during a mission.

Otherwise, by changing the preprocessing step also an adaptive pre-selection of the classifier can be carried out. Now before the actual classification task each image is compared to some appearance measures extracted from some images used to generate each dataset. Based on these similarity measures the—a priori—best classifiers can be

selected.
In addition and to obtain the more possible adaptive training dataset, another possibility offered by the offline approach is to use a certain amount of images taken from the input video to conduct an initial manual classification that successively becomes the training set and the classification may be then executed for the remaining (biggest) part of the video.
As final remark, note that during the offline processing, in parallel to the classification task also a more specific object recognition algorithm might be also executed.

2.3.3 Implementation details

A first prototype of the proposed framework was initially developed in Matlab code for an easy and fast development and evaluation. After, this code has been implemented in C/C++ (by using also the OpenCV libraries) to provide a faster implementation and to allow the integration with the ROS (Robot Operating System) to be executed on the vehicle hardware. In particular is only the classification module that actually needs to run on the AUV hardware, while the training phase may be achieved in a (sufficiently powerful) desktop PC.

Chapter 3

A method for underwater dehazing

Acquiring clear images in underwater environment is a key issue in ocean engineering [51]. Light-ray scattering and colour changes, usually lead to contrast loss and colour distortions in images acquired in underwater.
Conventionally the classic approach mostly rely on compensating both these issues with techniques more close to image enhancements or traditional histogram equalizations.
In this chapter is presented a method that try to recover the actual image radiance in underwater environment. This is a techniques referred as *dehazing* and substantially borrowed from the terrestrial scenario.
In the following sections the general problem is presented, both in the terrestrial and underwater scenario. After giving some backgrounds, in section 3.2 is presented the approach that likely is the most adopted method of image dehazing from single image that is applied to terrestrial images.
In section 3.3 the haze removal problem is extended to the underwater scenario, firstly analysing issues and some existing techniques. Then, in section 3.4 is discussed our proposed method for underwater dehazing and it is, followed by obtained results. At last, in section 3.6 the relationship between haze and image depth is investigated, showing how starting from the haze effect it is possible to infer a coarse 3D of the scene directly from a single image.

3.1 Introduction and general problem definition

During transmission from a point of a given scene to an observer, light rays can be affected by various degradations. *Scattering* and *absorption* are the major responsible for decreasing the quality of perceived image. As we saw in detail in chapter 1, light passing through different medium is deviated from its theoretically straight trajectory. Unless we have to deal with the geometry of image, this phenomenon, macroscopically

FIGURE 3.1: An everyday example of image distortion due to different medium interfaces.

is not a big deal for what concerns image quality and clearness (Figure 3.1). In this sense the distortion effects are mostly due to the number of crossed media.
Problem arises when we consider the microscopic deviations caused by non-homogeneous medium. The light that we perceive is the result of deviations and reflections of rays emitted by a source. In a normal clear day, sunlight is reflected by terrestrial object and hits our retina travelling through the air. In a foggy day we can perceive only closer object while those that are far away appear unfocused and faded (Fig. 3.2). This

FIGURE 3.2: Image characterized by fog (image from flickr).

phenomenon is caused by suspended particles (composed by water in this case) that scatter light rays. Depending on size and density of particle distribution the degradation increases. When the amount of particles is high the scene might be irremediably

occluded and not much can be done to recover images. From the other side, when the phenomenon is not so strong and some ray can travel from the object to the perceiver it is possible to get back some important details with appropriate techniques.
Particles can be of different nature than water, like dust or smoke, but anyway we refer it generally as *haze*.
Following the literature, the term *dehazing* refers to the techniques for haze removal. Even if they can appear sometimes similar, removing haze from images is different to other de-noising techniques based on classical image filtering. Differently from the acquisition noise, the haze effect has a natural origin, depending on scenarios and a light transmission model is needed.
After discussing the existing dehazing techniques and showing results of our implementation for terrestrial images, in this chapter we treat the case of underwater dehazing. In comparison of the air, the underwater haze has similar effects but some different causes that have to be properly treated.

3.2 Background

Dehazing techniques found their origin in the terrestrial environment. The haze reduces contrast, make colour grayer and objects are difficult to identify. Removing haze can increase the visibility of a scene and correct the possible colour distortions caused by the atmospheric lighting.
In this work the attention is focused on *single image* haze removal, but this is nothing than one of the last approaches to address this problem. The context of computational photography was the first interested in deal with haze and after that an increasing amount of attention was received also by the field of signal, and in particular, image processing. Several proposed methods in literature address this problem primarily with techniques close to the classic photography, like varying some camera settings during image acquisition.
Satellite imaging, otherwise, is one of the early fields (back to 1970s) that studied how to improve or restore the information in an image heavily characterized by haze. Atmospheric corrections and radiometric calibrations from satellite, largely employ a technique known as *dark object subtraction*, which basically consist in using as reference values those points that correspond to the darkest object in the scene ([64]). The satellite or airplane [65] imaging is not—even if represent a large number of applications—the only fields in which this problem is studied. Haze removal techniques are strictly connected also to coarse depth recovering and blur estimation ([66]).
Although, in general, haze removal from single image is based on a model and some a

priori knowledge of the atmospheric scattering, as it will be shown in the next section, this remain an ill-posed problem because the *airlight-albedo* ambiguity.
In one of the first major works about this problem ([2]) it has been observed by a statistical point of view, that the contrast in a clear image is considerably higher than in one affected by haze.
A simplified and widely used optical model of image formation is depicted in Figure 3.3. Due to atmosphere, the direct transmission of light that originates from the object, is

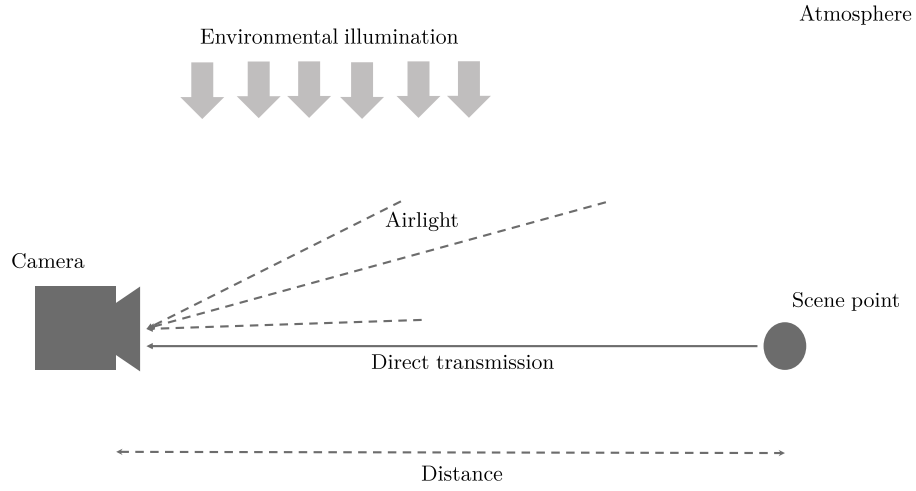


FIGURE 3.3: Optical model formation of an (hazy) image. (Image from [2])

influenced by other rays scattered away by suspended particles. The total ray-beam that hits the image sensor is then composed by two mixed components: the object-reflected rays and those scattered by atmosphere. This latter effect is the origin of *airlight*.
Clearly the actual amount of haze is strictly related to the combined effect between these two components. Several object reflections and the airlight methods from single image require to know the *airlight* parameter.
Starting from an acquired image, the airlight value may be provided by hand (generally indicating a sky region in image) or estimated automatically as for example proposed in [67].
Mathematically, the single image dehazing approach proposed by Tan in [2] starts from the classic optical model discussed in [68],[69] and commonly used in image processing and computer vision when dealing with light propagation in a given medium.
This model is:

$$I(x) = L_\infty \rho(x) e^{-\beta d(x)} + L_\infty (1 - e^{-\beta d(x)}) \tag{3.1}$$

where $I(x)$ is the image intensity value at pixel x and L_∞ is the atmospheric light (i.e. the airlight). It can be seen that the airlight is supposed constant in the whole image;

in fact for terrestrial images it does not depend in general on pixel position, but it is assumed as like it comes from infinity and its behaviour may be considered uniform. The other parameters in equation 3.1 are $\rho(x)$ that is the inherent reflectance coefficient of an object in the image, $d(x)$ or the sensor-object distance (i.e. depth) and β the attenuation coefficient. Considering linear optics, this latter coefficient derives from the general *extinction coefficient* [70], both handling the scattering and absorption phenomena. In practical applications, supposing uniform the atmospheric medium, also β may be kept constant in the model.

Clearly this is a simplified model, that does not consider microscopic light-particle interactions. Anyhow the importance of this simplified optical model is that, as we'll see, it has been employed with success across many other works, sometimes adopting different notation or additional assumptions.

Another early popular method, contemporary of this latter, is the one proposed by Fattal in [5]. The baseline idea is practically the same. This is again a *passive* method—in fact does not use any further specialized hardware than the camera—and basically it uses haze information and airlight estimation to recover a transmission map that will be successively used to restore the clearer appearance of the image.

Another method to jointly estimates the real scene reflectivity and depth starting from a single image is proposed in [3]. Here the underlying hypothesis is that both albedo and depth can be treated as two conditionally statistically independent image layers and authors use a Bayesian approach to model them. An hazy image is modelled as a Factorial Markov Random Field, where both chromaticity $C(x, y)$ and depth (i.e. the distance) $D(x, y)$ are the two hidden layers associated to the observations of the intensity image values at pixel $\boldsymbol{x} = (x, y)$ as reported in Figure 3.4.

The optical model used is practically the same than in equation 3.1 and assuming to know L_∞ it straightforwardly leads to a sum of the two terms, $C(x, y)$ and $D(x, y)$.

Largely inspired by previously discussed works ([2] [5]) is the method proposed by He in [4]. Today, this is likely the most influencing dehazing approach, especially in practical applications, because its relative simplicity and effectiveness. A wide amount of successive haze removal algorithms were developed starting from it and for this reason the section 3.2.1 will be entirely dedicated to analyse it.

Many current works based on He et al.'s method, suggest improvements in airlight estimation ([71]), in speed boosting ([72] [37],[73]) or in post processing phases ([74]). Obviously the haze removal can be extended to deal with also video sequences ([75] [76]). In those case the *single image* requirement might be relaxed and other techniques based on contrast enhancements may be used.

In [77] authors propose—somehow differently from previously described approaches—that instead to only look at the statistical aspects of dehazing, it is possible to start

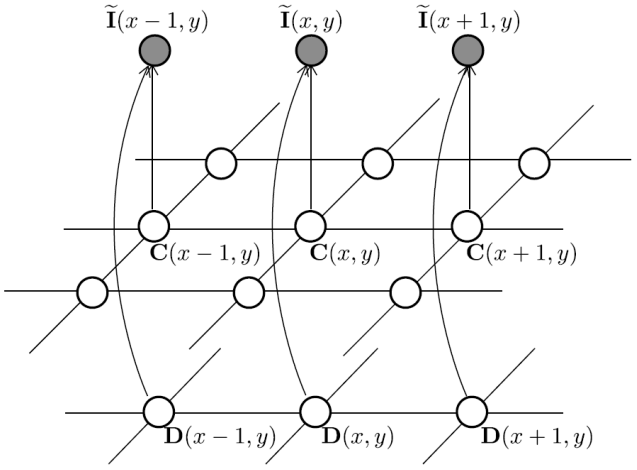


FIGURE 3.4: Markow Random Field formulation of the problem of albedo-depth joint estimation. $\widetilde{I}(x,y)$ is the known image pixel values, $C(x,y)$ is the albedo and $D(x,y)$ the relative depth. Image from [3].

from a geometrical point of view. In particular they made the substantially assumption of an image taken outdoor and dominated geometrically by a ground plane.

In certain applications, haze might also be treated as a noise reduction problem, considering its natural origin. In [78], for example, a method is shown from a single image to infer the noise as a function of image intensity. They uses a simple prior model for noise estimation without the knowledge of the image content.

To complete this short review on dehazing techniques based on single image, other methods, might be based on general contrast enhancement techniques. In particular they adopted *pixel-wise* operators that do not depend on spatial pixel relations. Examples are histogram equalizations, stretching, linear mapping and more complex tone reproduction operators ([79]).

Starting from the hypothesis that the haze, affect lower frequencies first, other methods (e.g. [80]), work with wavelets, use spatial pyramids, apply some bilateral filters (e.g.[81]) or gradient analysis (e.g.[82]) to preserve image edges.

Not all the existing methods for haze removal are based on single image post-processing techniques ([83]). In some cases solutions that use hardware components may alleviate the problem of haze before the image acquisition; as in the case of using polarized optical filters [84]. The light polarization changes when light rays are reflected or refracted by the interface between two different types of transmission medium. The main assumption is that the polarization of received light is almost caused by the environmental illumination, and only few beams, with more defined polarization (and typically lower intensity) are the only that really matter.

Other dehazing techniques are instead based on additional information coming from

other sensors or human interaction (e.g.[85]). Information about depth, even if it is not precise, can highly improve haze removal and image restoration ([86]).
For analogous reasons also a multi-image system may be useful; a stereo system, for example, can both provide depth information and point correspondences, and also any pair of blurred and noisy images can be sometimes sufficient, as proposed in [87].

3.2.1 He et al.'s method

As we said in the previous section, haze is an important cue to estimate the depth of a scene and vice versa, because their mutual dependence. From single image and without further information, the problem of dehazing results in general under-constrained. In particular to estimate the transmission—that formally is the portion of the light beam that starting from a target object can reach the camera sensor—we need to know the (relative) scene depth. The formula for the transmission $t(x)$ can be expressed as:

$$t(x) = e^{-\beta d(x)} \tag{3.2}$$

where β is defined as the scattering coefficient (see section 1.2) of the medium and $d(x)$ is the depth of pixel x. Basically, there is an exponential attenuation of light transmission with the depth.
Considering the RGB model He et al. in [4], after some statistical observations on images, pointed out that in most local image regions (with very few exceptions) there are some pixels with very low intensity values in almost one colour channel. Their idea is to look at these pixels in a similar manner than in *Dark Object Subtraction* methods, obtaining what they call *Dark Channel Prior* (DCP).
In hazy images, the darkest pixels have an intensity value that can be entirely attributed to the airlight component; by locally sampling these pixel over the whole image an accurate estimation of the light transmission into the scene can be provided.
Limitations in using this approach are only due to scenes where the entire image has an appearance almost equal to the airlight. An example is the sky, where pixel appearance is entirely due to the airlight and sky points are practically at infinity with all their appearance determined only by airlight.
The starting model is basically the same as reported in equation 3.1, with just some notation changes. The process of image formation ([88]), at pixel x, is described by the equation:

$$\boldsymbol{I}(x) = \boldsymbol{J}(x)t(x) + \boldsymbol{A}(1 - t(x))) \tag{3.3}$$

where $\boldsymbol{I}(x)$ is the observed intensity at pixel x, $\boldsymbol{J}(x)$ is the actual scene radiance, $\boldsymbol{A}$ is the atmospheric light and $t(x)$ is the transmission. It has to be noticed that $\boldsymbol{I}$, $\boldsymbol{J}$ and $\boldsymbol{A}$, include the three channel RGB representation.
Splitting the equation 3.3 we can identify two terms: the *direct attenuation* $\boldsymbol{J}(x)t(x)$ and $\boldsymbol{A}(1-t(x))$ that is properly the airlight component.
The transmission, intended as the portion of light that isn't scattered by the medium, is independent from the particular channel. Hence the transmission can be expressed as:

$$t(x) = \frac{I^c(x) - A^c}{J^c(x) - A^c} \quad c \in \{r, g, b\} \quad . \tag{3.4}$$

From this latter equation (3.4), assuming the dark channel prior hypothesis, for each local image patch $\Omega(x)$ centered on a pixel x, there must be at least one pixel with a very low value in almost one colour channel.
The map of all these pixel gives rise to the *dark channel* map, defined as:

$$J^{dc}(x) = \min_{y \in \Omega(x)} (\min_{c \in \{r,g,b\}} J^c(y)) \quad . \tag{3.5}$$

By considering previous hypothesis, for these pixel the actual (single channel) radiance may be considered zero, so:

$$J^{dc}(x) \to 0 \quad . \tag{3.6}$$

For common (natural) images it may be observed that depth and medium composition are not abruptly varying, so it can be done the hypothesis that every image patch $\Omega(x)$ has a constant transmission $t^{\Omega(x)}(x)$. Dividing the equation 3.3 by A^c, it may be rewritten as:

$$\frac{I^c(x)}{A^c} = t^{\Omega(x)}(x)\frac{J^c(x)}{A^c} - t^{\Omega(x)}(x) + 1 \quad c \in \{r, g, b\} \quad . \tag{3.7}$$

Following the dark channel definition (considering that A^c is not negative):

$$J^{dc}(x) = \min_{y \in \Omega(x)} (\min_{c \in \{r,g,b\}} \frac{J^c(y)}{A^c}) = 0 \quad , \tag{3.8}$$

and applying the *min* operator on both sides of equation 3.7,

$$\min_{y \in \Omega(x)} (\min_{c \in \{r,g,b\}} \frac{I^c(y)}{A^c}) = t^{\Omega(x)}(x) \min_{y \in \Omega(x)} (\min_{c \in \{r,g,b\}} \frac{J^c(y)}{A^c}) - t^{\Omega(x)}(x) + 1 \tag{3.9}$$

we finally obtain that the transmission $t(x)$ estimated for every patch is:

$$t(x) = t^{\Omega(x)}(x) = 1 - \min_{y \in \Omega(x)} (\min_{c \in \{r,g,b\}} \frac{I^c(y)}{A^c}) \quad . \tag{3.10}$$

Once obtained an estimate for the whole image transmission, the actual radiance can be recovered by the equation (simply derived from the equation 3.3):

$$\boldsymbol{J}(x) = \frac{\boldsymbol{I}(x) - \boldsymbol{A}}{\max(t_0, t(x))} + \boldsymbol{A} \tag{3.11}$$

where a minimum threshold t_0 for the transmission is used to avoid issues occurring when the estimated $t(x)$ is close to zero.
Until now, nothing has been said about the $\boldsymbol{A}$ parameter. It was supposed known, but actually it needs to be estimated. Instead follow the approach proposed in [2]—that simply takes the brightest pixel in image—He et al.'s method again uses the *dark channel* map; in fact, this can be regarded as an approximation of the haze density, so they pick up as atmospheric light $\boldsymbol{A}$ the brightest pixels intensity value taken from a set including only a small percentage (typically 10%) of the highest values in the dark channel image.
Furthermore, to conclude this description we need two final remarks:

- The He et al.'s single image dehazing method considers to slight modify the equation in 3.10 for the transmission, adding a constant parameter $\omega \in (0, 1)$ to keep a little amount of haze. This gives a more natural appearance to the final recovered images, hence the actual transmission equation is:

$$t(x) = 1 - \omega(\min_{y \in \Omega(x)} (\min_{c \in \{r,g,b\}} \frac{I^c(y)}{A^c})) \quad . \tag{3.12}$$

- Transmission evaluation is made by considering small squared image window where this value is assumed constant. This may lead to a non pleasant final appearance, with squared appreciable artificial patches making the image unnatural. To overcome this effect, before recovering the radiance (last step) the obtained transmission map is processed by a refinement algorithm that smooth the transition between patches. This is achieved by a *soft matting* algorithm ([89]) plus a bilateral filtering to smooth and preserve edges.

In general we can observe that both the refinement step and the atmospheric light estimation are those that differentiate the largest amount of dehazing methods from the previous. The refinement step can hardly degrade the performance in terms of required time, so depending on the particular applications may be avoided or changed with faster (and coarser) approaches, even if this often means to renounce to some detail.
In Figure 3.5 are reported some results of experiments that we achieved by implementing the He et al. method (Matlab® code). On the left column are reported the original

FIGURE 3.5: Some examples of dehazing. Original starting images are in the left column and their recovered version is on the right in the same row. Center column reports the transmission $t(x)$ computed, without refinement (source images derive from [4], [5] and [2])).

starting images, on the middle the (non-refined) transmission while the actual image recovered are in the third column.

It can be observed as the dehazing algorithm is actually capable to give back to the image more tone, sharpness and increasing the visibility of smaller details.

3.3 Underwater dehazing

More than the air, water is an hard opponent to the light transmission. Haze induces poor visibility in terrestrial atmosphere, but even more in underwater environments

where, in general, suspended particles are greater in number and cause an high scattering.

Differently from the air-medium, water presents strong effects also on absorption. Both phenomena—scattering and absorption—can be proved also in human experience.

The field of view is shorter (also extremely short in some kinds of lakes or rivers) and colours are notably distorted, shifted to a bluer or greener appearance. This latter fact is due to the variable attenuation of frequencies in water. Furthermore as shown in Figure 3.6 the red colour is usually characterized by less intensity (i.e. power) when it reaches the camera sensor in comparison to the other RGB channels. Sometimes it can be very

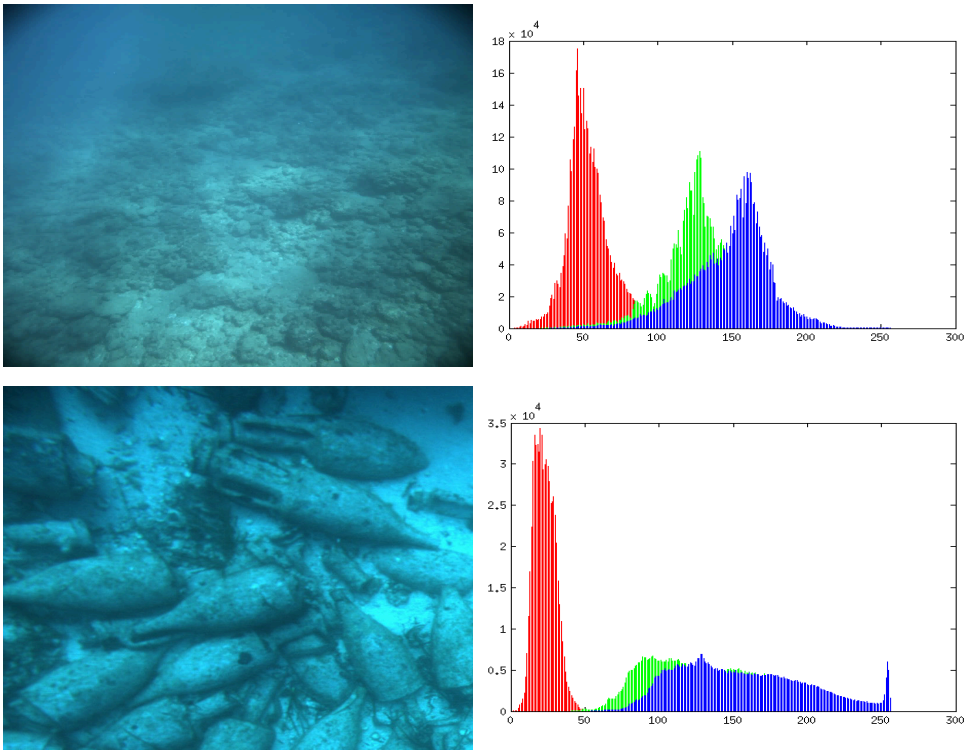


FIGURE 3.6: Underwater images (left column) with their corresponding RGB histogram plot (right column). It is evidenced how the red channel is globally at lower values than blue and green channels (images from ARROWS project).

low, but this does not mean that it has not information at all.

In this situation the He's et al.'s method might be useless because the Dark Channel Prior tends to overlap with the red intensity values and it is a poor indicator about the actual haze effect distribution.

Here we are interested to single image haze removal methods, but as well as the terrestrial scenario, these are not the only suitable approaches for radiance recovering. Additional informations might be employed in a similar manner than in air-medium.

The problem of single image dehazing in underwater environment may be formulated in a close manner; even if the effects are common, anyway there is a variation in causes that generate them.
To our knowledge the majority of approaches (not so much, really) for underwater dehazing are less or more based on the He et al.'s method and its prior (DCP).
As stated in Duntley's work [16] the full general underwater model can be actually considered:

$$N_t(z,\theta,\phi) = N_{t0}(z_t,\theta,\phi)e^{-\alpha(z)r} + N(z_t,\theta,\phi)e^{k(z,\theta,\phi)r\cos\theta}(1 - e^{-\alpha(z)r+k(z,\theta,\phi)r\cos\theta}) \tag{3.13}$$

where:
$N_t(z,\theta,\phi)$ is the observed radiance,
z is the depth (observer),
z_t is the depth (target),
θ is the zenith (observer-target),
ϕ is the azimuth (observer-target),
r is the observer-target distance,
$N_{t0}(z_t,\theta,\phi)$ is the actual radiance,
$N(z_t,\theta,\phi)$ is the radiance in the water column (the airlight),
$\alpha(z)$ is the attenuation rate,
$k(z,\theta,\phi)$ is the radiance attenuation function that captures how the airlight changes with z.
This model is quite complex enough due to the high number of parameters and function that include. With good assumptions that are valid in the majority of cases—in particular $\theta \approx \pi/2$ and constant α—the model become:

$$N_t(z,\theta,\phi) = N_{t0}(z_t,\theta,\phi)e^{-\alpha r} + N(z_t,\theta,\phi)(1 - e^{-\alpha r}) \quad . \tag{3.14}$$

This latter equation can be then reported, with appropriate notation changes, to the wide-used known model of equation 3.3. This justify the validity of such a model also for the underwater scenario.
Differently from the terrestrial case, what has to be considered in underwater environment is that the attenuation rate α is made by two component, respectively the scattering (α_s) and absorption (α_a) summed together ($\alpha = \alpha_a + \alpha_s$). In the terrestrial case solely the scattering component is considered; however, using the absorption component does not change substantially the model equation.
The most straightforward application of the theory of terrestrial DCP to underwater images is reported in [90]. Here authors practically leave unchanged the He et al.'s

method and apply it both in natural and in artificial underwater images.
A more general and analytic treatment is the one reported in [91], where a region-specialized method for underwater images is proposed with the aim to dehaze and compensate colours together. Another similar approach with DCP to handle haze and colour distortion simultaneously is proposed in [92].
Different from these methods that leaves the DCP practically unchanged with only few or none modifications, are the approaches proposed in: 1) Carlevaris-Bianco et al.[93], 2) Drews et al.[94] and 3) Wen et al.[6]. Still continuing to use the same DCP algorithm scheme, they include into their models proper characteristics related to the underwater environment; in particular all these methods starts from changing the prior and tacking in deeper consideration the underwater peculiarities.
More in detail:

1. In Carlevaris-Bianco et al.'s work, authors start from the dark channel prior approach but they modify this prior on the basis of the assumption that the red channel versus blue and green ones has, in underwater, a particular behaviour. They use a prior calculated as:

$$D(x) = \max_{x\in\Omega(x),c\in\{red\}} I^c(x) - \max_{x\in\Omega(x),c\in\{blue,green\}} I^c(x) \tag{3.15}$$

that means to take the difference between red and green-blue channels.
To estimate the transmission over every patch $t^{\Omega(x)}$, it is assumed that the closest foreground pixel has a maximum difference of one by normalizing all values in $[0, 1]$. Hence, the actual transmission is computed as:

$$t^{\Omega(x)} = D(x) + (1 - \max_x D(x)) \quad . \tag{3.16}$$

Starting from this transmission equation the radiance recovering step is carried out considering the usual model obtained from equation 3.3 (for all the three RGB channels):

$$\boldsymbol{J}(x) = \frac{\boldsymbol{I}(x) - \boldsymbol{A}}{t^{\Omega(x)}(x)} + \boldsymbol{A} \tag{3.17}$$

and successively modelling it as a noisy variable,

$$\boldsymbol{J}(x) = \boldsymbol{J}_0(x) + w(x) \quad . \tag{3.18}$$

Here, $\boldsymbol{J}_0(x)$ is the actual radiance value and $w(x)$ is a white Gaussian noise ($w \sim \mathcal{N}(0, 1)$). In this way the final radiance values can be computed as a *maximum a*

posteriori estimate, maximizing the posterior probability:

$$P(\boldsymbol{J}_0(x)|\boldsymbol{J}(x)) \propto P(\boldsymbol{J}(x)|\boldsymbol{J}_0(x))P(\boldsymbol{J}_0(x)) \quad . \tag{3.19}$$

Clearly this last step of considering $\boldsymbol{J}(x)$ as a noisy process, is an optional upgrade of the original (terrestrial) dark channel prior based method.
For what concerns the atmospheric light, the $\boldsymbol{A}$ value is calculated as:

$$\boldsymbol{A} = I(y) \quad with \quad y = \arg\min_x t^{\Omega(x)}(x) \tag{3.20}$$

that practically is the RGB values corresponding to the lowest transmission pixel. It might be pointed out that—mostly when the underwater images are taken with a camera with the principal axis perpendicular to the seabed plane—these airlight points are expected to be located on pixels that correspond to furthest scene points.

2. In Drews et al.'s work the baseline assumption is easy. In underwater environment the red channel is affected by stronger absorption than the other two RGB channels, so the idea is to extend (or limiting) the original DCP approach to deal only with the green and blue channels.
   It results in a slightly new prior that the authors call *underwater DCP* ($J^{udcp}(x)$) and computed as:
   $$J^{udcp} = \min_{y\in\Omega\{x\}}(\min_{c\in\{g,b\}}(J^c(y))) \tag{3.21}$$
   Aside this, the other steps of this method to restore the actual image radiance actually remain the same as in He et al.'s work. The constant parameter $\boldsymbol{A}$ is directly computed tacking the RGB values correspondent to the brightest pixel among those in $J^{udcp}(x)$.

3. In Wen et al.'s work, is taken again into consideration the different behaviour of red channel—differently from blue and green—characterizing the underwater environment.
   Starting from the usual (hazy) image formation model (3.3) here the assumption is to employ different transmission maps for each colour channel. In particular two transmission are considered: one for the red channel and another (identical) for blue and green. Then, they model the underwater image formation process as:
   $$I^c(x) = J^c(x)t^c_\beta(x) + B^c t_\alpha(x) \tag{3.22}$$
   where $c \in \{red, green, blue\}$ and B^c is the *background light* that substantially has the same role played by the atmospheric light $\boldsymbol{A}$. It has to be noticed that the

transmission is now expressed by two different component $t_\alpha(x)$ and $t^c_\beta(x)$. The first, t_α includes only the scattering effect and it is the same independently from the colour channel. Instead, t^c_β is dependent from the channel and try to catch both effects of absorption and scattering. The separation in two different transmission is due to the consideration that the absorption phenomenon only affects the amount of light originating from the target object and can be neglected in the airlight component.

The prior t_α is computed minimizing only on the green and blue channels. Dividing the equation 3.22 by B^c—similarly than in the He at al.'s method— is obtained the equation:

$$\min_{c\in\{b,g\}}(\min_{y\in\Omega(x)}(\frac{I^c(y)}{B^c})) = t_\beta^{\Omega(x)} \cdot \min_{c\in\{b,g\}}(\min_{y\in\Omega(x)}(\frac{J^c(y)}{B^c})) + t_\alpha^{\Omega(x)} \tag{3.23}$$

where both transmission $(t_\beta^{\Omega(x)}, t_\alpha^{\Omega(x)})$ are assumed constant in every patch $\Omega(x)$. Under the hypothesis that the dark channel in equation 3.21 tends to zero—in fact the B^c parameter is positive and constant—it results that the scattering transmission $t_\alpha(x)$[1] is:

$$t_\alpha(x) = \min_{c\in\{g,b\}}(\min_{y\in\Omega(x)}(\frac{I^c(y)}{B^c})) \quad . \tag{3.24}$$

Otherwise, for blue and green channels, the transmission $t_\beta(x)$ can be obtained as:

$$t^{g,b}_\beta(x) = 1 - t_\alpha(x) \tag{3.25}$$

and, instead, the transmission for the red channel, t^r_β as:

$$t^r_\beta(x) = \tau \cdot \max_{y\in\Omega(x)} I^{red}(y) \tag{3.26}$$

where τ is a correction parameter to normalize this transmission value.

The radiance recovering for each channel c, can finally be performed by the equation (derived from 3.22):

$$J^c(x) = \frac{I^c(x) - B^c \cdot t_\alpha(x)}{t^c_\beta(x)} \quad . \tag{3.27}$$

To apply this latter formula $\boldsymbol{B} = (B^{red}, B^{green}, B^{blue})$ has to be firstly estimated. To perform this step they use the pixel value at the image position P such that:

$$P = \arg\min_x(I^{dark(red)}(x) - \max(I^{dark(green)}(x), I^{dark(blue)}(x))) \tag{3.28}$$

[1]For a more readability in the following we stop to report the apex $^{\Omega(x)}$ unless when strictly needed.

with:

$$I^{dark(c)}(x) = \min_{y \in \Omega(x)} (I^c(x)) \quad . \tag{3.29}$$

Summarizing, to overcome the red-light attenuation, Drew et al.'s and Wen et al.'s method both apply the DCP theory, but limited only to the blue and green channel. This choice allows to better capture the underwater haze caused by the scattering effect, even if sometimes they tend to overestimate it.
Instead, the Carlevaris-Bianco et al.'s approach is based on a different prior (eq. 3.16) and uses Markov Random Fields to better recover the actual image appearance. Despite its good performance—mostly accomplished with foreground objects—it suffer the presence of artificial illumination and strongly depends on the chosen airlight value (sometimes the human interaction is preferable). As the Drews et al.'s method, the Bianco's one does not handle directly the absorption but only the scattering.
Among these three approaches presented, only the one proposed by Wen et al. ([6]) takes into account this phenomenon. Even if the absorption is not directly modelled, it has made a channel distinction (equation 3.22). Anyhow it does not resolve all the illumination problems, but we tested that in general, all the presented methods suffer problems related to non-uniform, low or artificial illumination. The method for underwater dehazing that we introduce in the following section is firstly aimed to overcome such limitations.

3.4 Underwater dehazing: proposed method

In proposing a different approach for underwater dehazing from single image, our goal is to improve—as much as possible—the performance of existing solutions. In particular we want to achieve higher independence in illumination and scenarios changes. Images on the right column of Figure 3.7 show how the illumination is distributed by considering two different scenarios (left column). The top-one, where the lighting is more irregular represents the sunlight effect while the bottom shows the case of artificial illumination, characterized by a more regular distribution. The architecture of our proposed method for single image underwater dehazing is fairly similar to the one shared also by other major work in this field. Figure 3.8 reports our adopted scheme. The fundamental blocks are those of *total transmission* and *airlight estimation*, other than the refinement process and the actual final step of radiance recovering. Except for the scattering evaluation, all the other blocks separately work over the three different RGB channels.

The basic assumption in our proposed method for underwater dehazing is that, differently from the terrestrial case, it needs to both model separately the effects due to

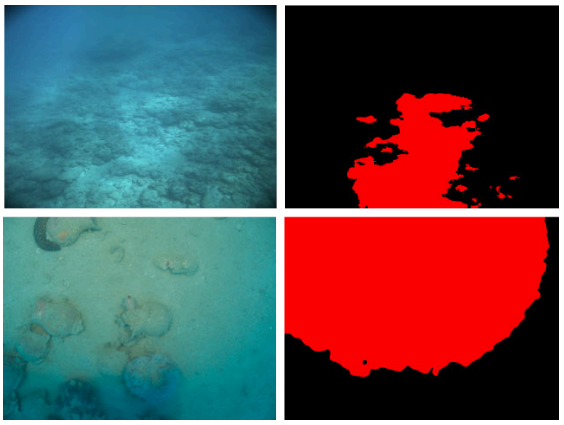


FIGURE 3.7: Two different scenarios about lighting distribution. In the top-left image the sunlight gives rise to a non uniform seabed illumination as underlined in the corresponding right figure. The bottom image shows the case of artificial illumination, that—also depending on the number of lights—can be detected by analysing the regularity of the illumination edges.

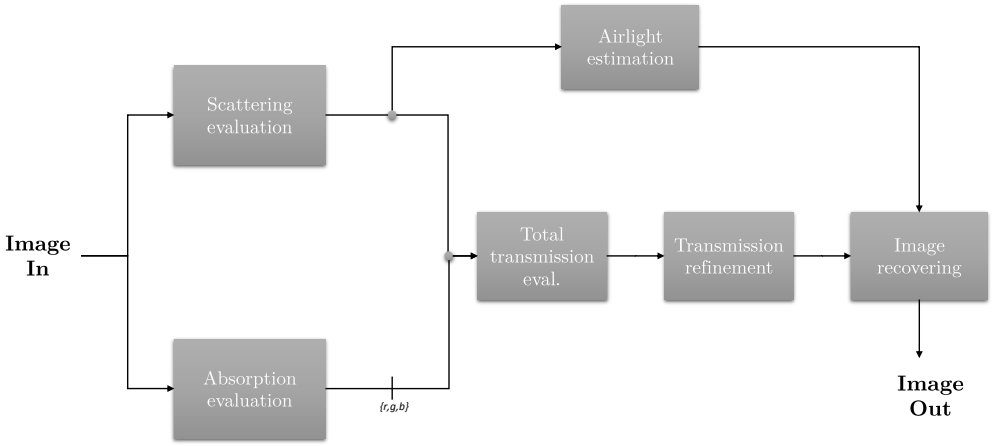


FIGURE 3.8: The architecture of our proposed method. Note that except for the scattering estimation (*haze*) the other blocks work in parallel with the three RGB channels and are computed over all colour channels and then merged in the last step.

scattering and absorption.

We previously saw that the full Duntley's equation (3.13), completely describing the light transmission model in an underwater scenario, might be simplified in a form that is fairly close to the usual model,

$$\boldsymbol{I}(x) = \boldsymbol{J}(x)t(x) + \boldsymbol{A}(1 - t(x))) \tag{3.30}$$

where the bold letters represent vectors or matrices on the RGB channels (this equation with a deeper description was already presented in section 3.3). This correspondence leads to define the total transmission as:

$$t(x) = e^{\alpha} = e^{\alpha_s} e^{\alpha_a} \tag{3.31}$$

where α_s and α_a are respectively the scattering and absorption coefficients.
The knowledge of such coefficients or an empirical estimation of them (see also chapter 1) makes possible to recover the actual image radiance $\boldsymbol{J}(x)$ from $\boldsymbol{I}(x)$.
Despite the impossibility to directly estimate α_s and α_a from a single image our method considers as separate the two effects and models the total transmission $\boldsymbol{t}(x)$ as composed by two single contributions $s(x)$ and $\boldsymbol{a}(x)$. By assuming for now their independence we can express the total transmission as:

$$\boldsymbol{t}(x) = \boldsymbol{a}(x) s(x) \tag{3.32}$$

where $\boldsymbol{t}(x)$ and $\boldsymbol{a}(x)$ are vectors representing the quantities for the three colour channels. In general total transmission $t(x)$ is now seen as a function of this two varying measures. By comparison, in (the simpler) terrestrial scenario the only transmission actually considered is $t(x) = e^{\alpha_s} = s(x)$. In underwater environments the evaluation of both transmission components must be carried out in a separate way. In particular the scattering coefficient can be evaluated as the Dark Channel Prior method without considering the red channel. The transmission due to scattering may be expressed as:

$$s(x) = 1 - \min_{y \in \Omega(x)} (\min_{c \in \{g,b\}} \frac{I^c(y)}{A^c}) \tag{3.33}$$

where $\Omega(x)$ is a neighbourhood of the pixel x on image $I(x)$, A is the airlight and c represents the considered channel ($\{red, green, blue\}$). As we seen this is a common choice in literature to handle haze in the underwater environment because the red channel, in areas not close to the camera is fairly low.
The second factor in the total transmission model is the absorption (computed for each channel $c \in \{r, g, b\}$); it is evaluated as:

$$a^c(x) = \max_{y \in \Omega(x)} (\frac{I^c(y)}{A^c}) \quad . \tag{3.34}$$

Despite its simplicity this value can quickly give information about the light absorption caused by water medium. Although this quantity is evaluated by considering separately the single channels, sometimes we refer the transmission due to absorption only with $a(x)$ and indicating in this case the average (on the three channels) value. Clearly the

given representation for $\boldsymbol{a}(x)$ might have its failures in special areas, with particular uniform colour gradients or patterns that may confuse the absorption estimation. From the experience carried out by intensive tests, the underwater environment presents highly varied and irregular colour distributions that leads to neglect such particular situations, with using a sufficiently large neighbourhood window $\Omega(x)$. In any case, however, also employing more complex formulas, similar issues may still remain because none of them can resolve the inherently ambiguity behind the general absorption estimation from single image without the knowledge of the point distance.

Figure 3.9 shows an example of the scattering and absorption estimation for the underwater input images already shown in Figure 3.7. Images in column *(a)* and *(b)* represent

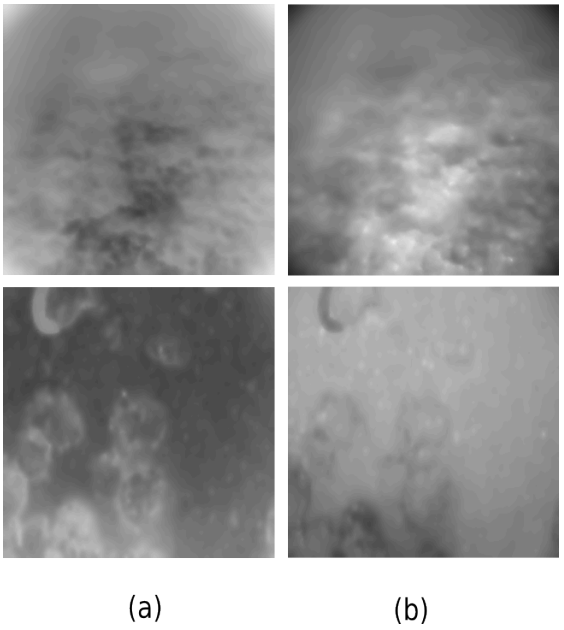


FIGURE 3.9: Examples of the $s(x)$ and $a(x)$ maps (the latter is averaged over RGB channels), reported respectively in column *a* and *b*. Components are both normalized in $[0, 1]$ and the neighbourhood window $\Omega(x)$ has a size of 15×15 pixels. Higher (white) intensities represents high transmission due to scattering effect $s(x)$ and due to absorption component $a(x)$.

examples of computation respectively of the $s(x)$ and $a(x)$ factors.

Scattering and absorption component are both normalized in $[0, 1]$ by the airlight component A^c, and the neighbourhood window $\Omega(x)$ has a size of 20×20 pixels. We notice that higher (whiter) intensities represents high transmission due to scattering ($s(x)$) and to absorption $\boldsymbol{a}(x)$ effect.

Evaluating only the scattering effect in underwater images might be trivial due the fact that the dark channel prior assumption is more weak than in the terrestrial environment. As we can see from the top left *(a)* image in Figure 3.9, considering only the scattering component in the underwater environment might lead to confuse higher illuminated areas with those that are the most haze-affected; a similar behaviour is shown also by

the lower image, where the illumination is artificial. Scattering and absorption might present an overlapped behaviour confirming the fact that these measures are not totally complementary or independent (it must be kept in mind that the quantity s(x) is the transmission due to scattering and not the amount of the scattering itself).
Using two components to express the total transmission function $t(x)$, there are four extremal and qualitative configurations that can be actually identified:

1. High scattering and low absorption phenomena ($s(x) \to 0$ and $a(x) \to 1$)
   This configuration represents the situation in which the haze is present and it is relatively close to the camera without significant absorption in all channels (also some saturated parts might show a similar behaviour).

2. Low scattering and low absorption phenomena ($s(x) \to 1$ and $a(x) \to 1$)
   This is the scenario in where less or no absorption and scattering are detected. Here the recovered radiance is substantially the same than in the input image.

3. High scattering and high absorption phenomena ($s(x) \to 0$ and $a(x) \to 0$)
   This scenarios is typical in points that are far away from the camera. Here, also in clear water, the haze is maximum and the hypothetical presence of objects is hard to recover.

4. Low scattering and high absorption phenomena ($s(x) \to 1$ and $a(x) \to 0$)
   Commonly this is a scenario represented by dark areas (i.e. areas with a good scattering transmission but total absorption).

The Figure 3.10 visually explains the previous four qualitative possibilities of $\boldsymbol{a}(x)$ and $s(x)$, that arise considering high/low scattering/absorption transmission.

Even if the total transmission is theoretically a multiplication of two factors $s(x)$ and $\boldsymbol{a}(x)$, respectively related to the scattering and absorption effects, in practice we cannot simply multiply them. Despite the correctness of such a representation, in this case the resulting total transmission would be lower than the actual value. This issue is linked to the fact that $s(x)$ and $\boldsymbol{a}(x)$ are evaluated in a non-independent manner; the transmission estimation due to scattering may also include a (un)certain amount of absorption in each colour channel and vice versa.
All these motivations lead us to actually employ a total transmission expressed as:

$$\boldsymbol{t}(x) = \max\left(s(x), \boldsymbol{a}(x)\right) \tag{3.35}$$

where $\boldsymbol{t}$ and $\boldsymbol{a}$ are both vectors with 3 (RGB) components. Remembering from equation 3.17 the radiance recovering formula, we have that the actual radiance $\boldsymbol{J}(x)$ is inversely

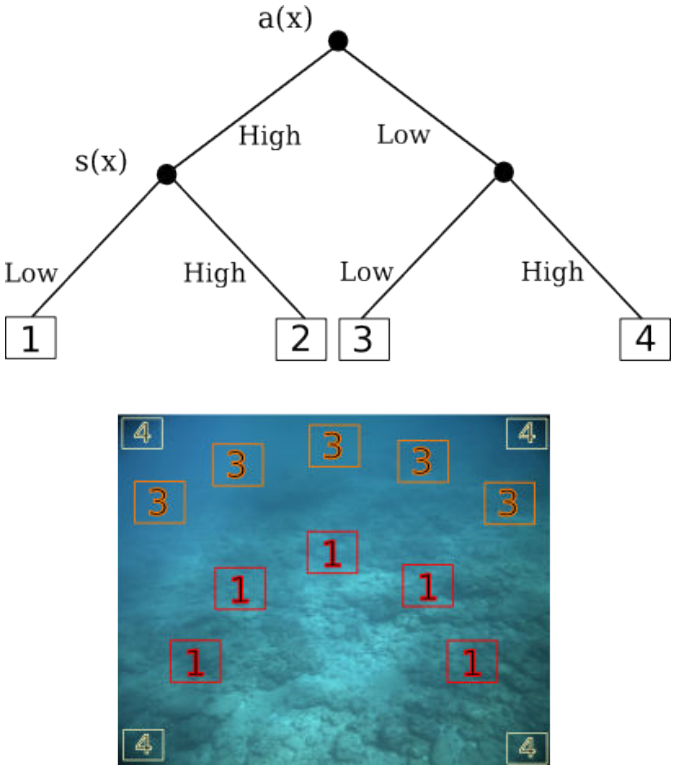


FIGURE 3.10: The four qualitative possibilities of $\boldsymbol{a}(x)$ and $s(x)$ synthesized with examples on a test image. Both $a(x)$ and $s(x)$ are continuously varying and mutual dependent. Areas labelled with number 2 are theoretically the clearest, but are also rare to find in underwater. The 1-labelled areas are characterized by an overall good visibility keeping low the absorption effect (high $\boldsymbol{a}(x)$). The increase of depth usually leads continuously to an augmented haze, as in label 3. Finally, the number 4 presents high $s(x)$ and lower $\boldsymbol{a}(x)$ that may be encountered in dark or far away areas in presence of a clear water medium.

proportional to total transmission $\boldsymbol{t}(x)$ which is in the interval $[0, 1]$. In particular lower $\boldsymbol{t}(x)$ means low (darker) output radiance and multiplying both $\boldsymbol{a}(x)$ and $s(x)$ generally leads to an underestimation of the total transmission. Otherwise, by choosing, in a pixel-wise fashion, the highest value between the two components we make an overestimation of the total transmission, and a brightest final radiance $\boldsymbol{J}(x)$ is obtained at the cost of keeping some degradation effect in the resulting image. An extensive amount of comparison experiments with other different transmission estimations have confirmed us the validity of this choice.

The airlight $\boldsymbol{A} = [A^r, A^g, A^b]$ is evaluated by only considering the scattering map value (i.e. $(1 - s(x))$) in a similar way than the He's terrestrial method. $\boldsymbol{A}$ is selected as the highest intensity RGB value over the entire original input image by choosing among those pixels that have bigger values on the direct scattering map (typically the set with the 10% biggest values is used).

As shown in Figure 3.8, representing the architecture of our method, once estimated $\boldsymbol{A}$ and before the $\boldsymbol{J}(x)$ computation, we filter the total transmission map to avoid the

block effect over the image caused by the window size used to locally estimate scattering and absorption contributions.

Both maps ($s(x)$ and $\boldsymbol{a}(x)$) are refined applying a filtering method similar to the one presented and improved respectively in [95] and [96]. In particular it is a*guided filter*, meaning that to refine our total transmission map we use a guidance, and in particular this guide is the input image itself. In comparison of other common image filters with explicit predefined kernels (e.g. *Sobel* and *LoG*), this guided filter better preserve edges of the input (guide) image. With a close behaviour in comparison to the *bilateral filter* ([97]), the guided filter is in general faster (especially using the *fast* implementation that employs sub-sampled images) and achieves comparable performance. By assuming a local linear dependence between the guide image (I) and the output image ($t = aI + b$), this filter is more than just a smoothing approach because allows to easily transfer the structure of the input image to the output. In our case this image is the total transmission map without the block-artefacts due to the scattering and absorption estimation. For this reason this technique is highly suggested for dehazing than the slower soft-matting approaches used in the earliest works. The Figure 3.11 shows an example of a transmission map before and after the refinement.

With this refinement step the output produced by the radiance recovering acquires a

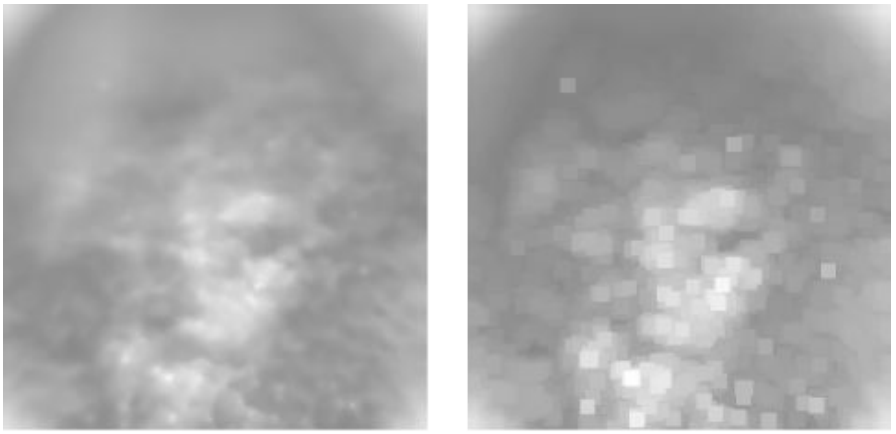


FIGURE 3.11: An example of a transmission map refined (a) and not (b). As can be noticed in the non-refined version are appreciable blocks corresponding to the window used to estimate the total transmission (including both $a(x)$ and $b(x)$).

more clear and pleasant look, without the block effects, as shown in Figure 3.12.

After the transmission map refinement, to finally obtain the expected radiance $\boldsymbol{J}(x)$ of the input image, a parameter $t_{low} \in (0, 1]$ needs to be introduced and the actual recovering equation takes the following form:

$$\boldsymbol{J}(x) = \frac{\boldsymbol{I}(x) - \boldsymbol{A}}{\max(t_{low}, \boldsymbol{t}(x))} + \boldsymbol{A} \tag{3.36}$$

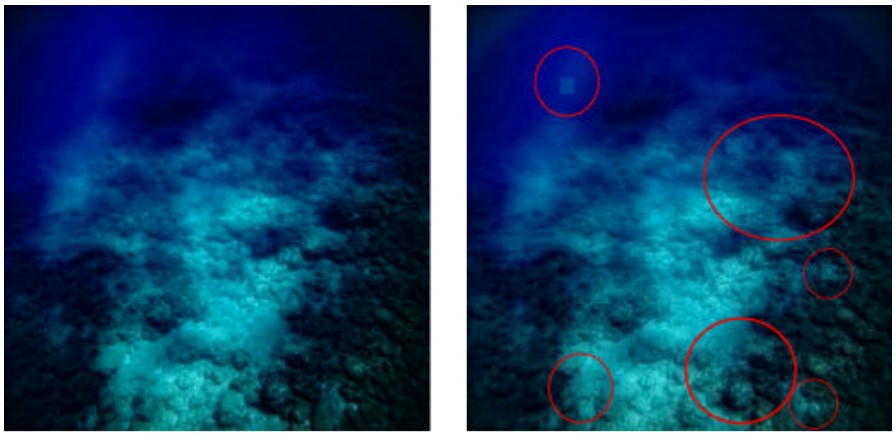


FIGURE 3.12: Radiance recovered considering the refined total transmission map *(a)* and the non-refined version *(b)*. Circles represent those areas where the block-effect on the output image is more evident.

where $\boldsymbol{t}(x)$ is defined as in 3.35 and t_{low} is needed to lower bound the amount of total transmission and keeping a certain amount of the input image $\boldsymbol{I}(x)$. Figure 3.13 shows an example obtained once the full dehazing process is completed and compares it with the other principal dehazing methods.

While He and Wen's methods both tends to introduce several artefacts on image, it is evident the close output between our method and the Drews's one. This similarity might occur, as in this case, when our total transmission approximatively follows the $s(x)$ component. The Drew's method represent in this way a subset of our approach. Stronger differences arise however when the absorption effect is widely less than the scattering, for example as in the Figure 3.14. From these preliminary comparisons it can be noticed also that our algorithm is much more stable using smaller lover bound for the transmission[2] (t_{low}) than the other underwater approaches. Obviously some methods are designed to perform better for certain situation than other (e.g. foreground and/or close objects).

3.4.1 Variable airlight

As can be seen in Figure 3.13, our proposed method might still produce dark areas in presence of non-uniform illumination. This phenomenon is due to the way for estimate the airlight, that is unique over the entire image. This phenomenon is also common in practically all methods and sometimes it may be hidden by choosing high lower bound for the transmission values. The actual problem that has to be handled is, instead,

[2]Some authors suggest to use bigger values to lower bound the underwater transmission, although increasing the t_{low} actually means having an output image that is closer to the input one.

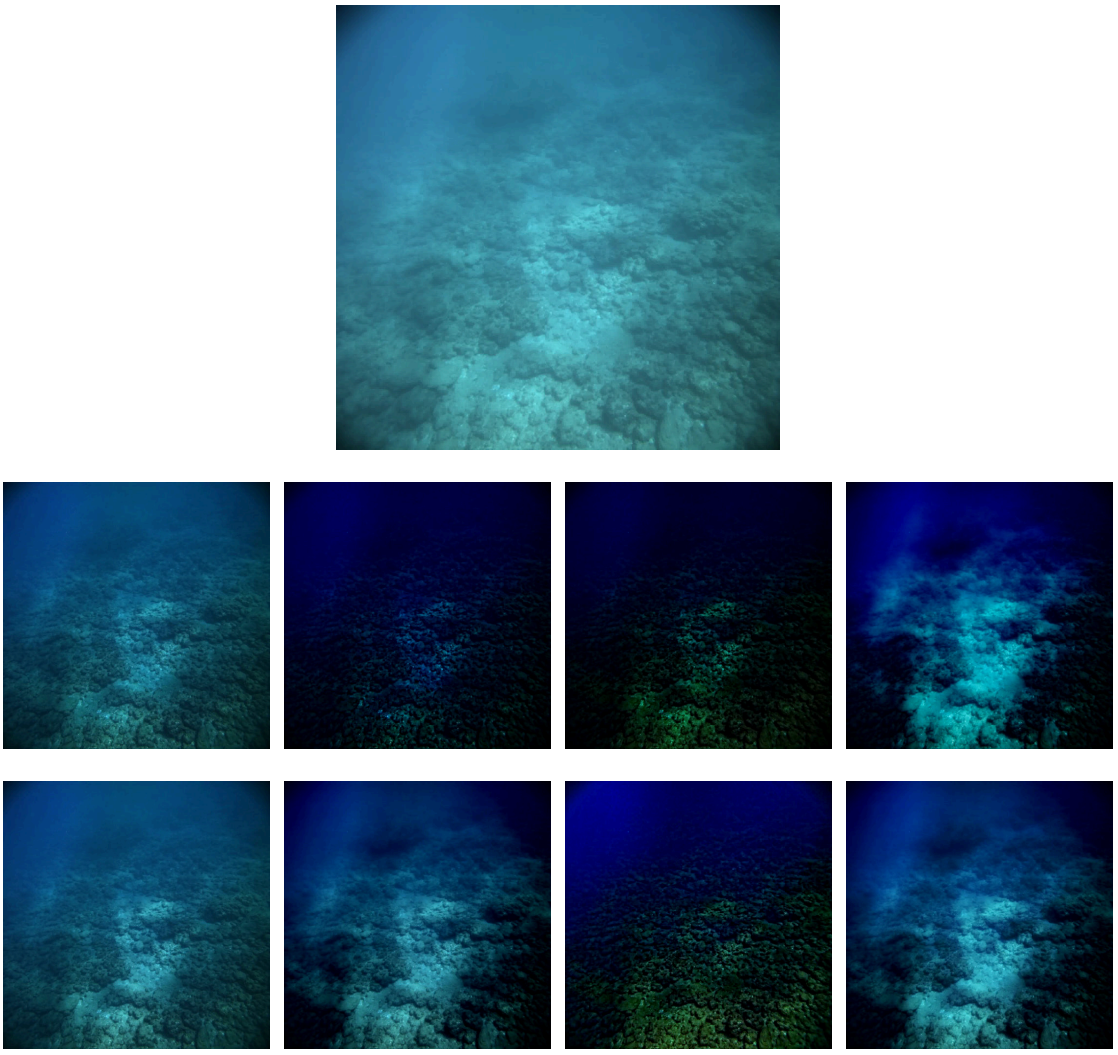


FIGURE 3.13: Results obtained with the main dehazing method in an underwater scenario. The single top image is the input, while the two rows corresponds to the same image with different transmission lower bounds, respectively $t_{low} = 0.2$ the first row and $t_{low} = 0.6$ the second. In both cases the processing was carried out with $\Omega(x) = 21 \times 21$ pixels and an original image of $2,5$ Mpx). By columns are reported the output images obtained with (starting from left): *1) He's method*, *2) Drews's method*, *3) Wen's method* and *4) our method*. It can be observed as to a lower values of t_{low} correspond in general a darker image. The He's method is the one that is not specifically designed for underwater and actually doesn't alter substantially the input image, while Wen's is the one that introduces more artefacts. Our method performs quite close to the one of Drew in this scenario but our methods appears notably less insensitive to the t_{low} values.

the non uniform illumination that might cause the alternation of bright and dark areas during the recovering of the actual radiance. More specifically, this is related to the airlight ($\boldsymbol{A}$) estimation. As well as for the medium absorption, usual terrestrial methods for dehazing does not handle this issue, or there are properly designed methods, as the one reported in [98]. Still keeping as reference the same architectural scheme reported in Figure 3.8, in our dehazing method we changed the way in which $\boldsymbol{A}$ is calculated by

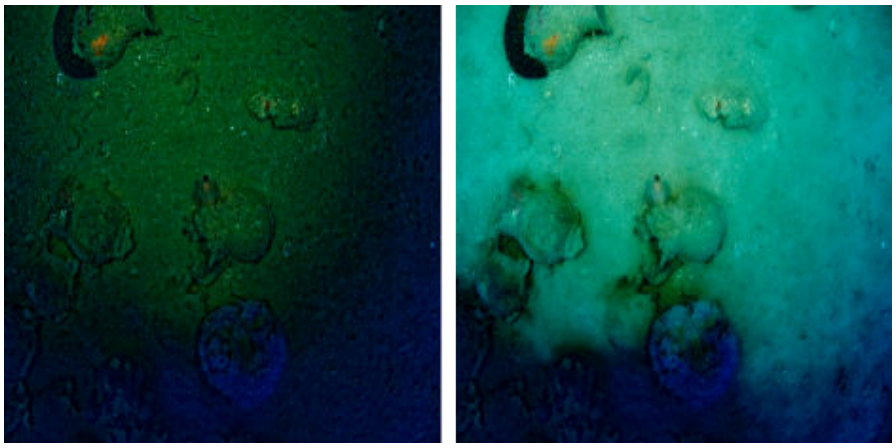


FIGURE 3.14: An image characterized by a strong artificial illumination recovered by the Drew's method (left image) and our approach (right image). These results were achieved with a limited $t_{low} = 0.3$ and confirm that the use of a measure of absorption in combination with one of scattering for the transmission estimation is a valid approach to face up to the underwater dehazing.

employing an adaptive approach.

The airlight is a three channel matrix $\boldsymbol{A}_v$, where each entry represents the local airlight value calculated over a $m \times m$ square patch in the input image. The size of the patch

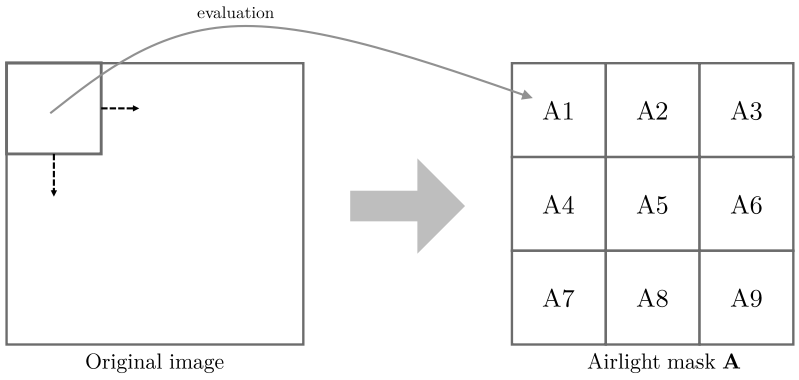
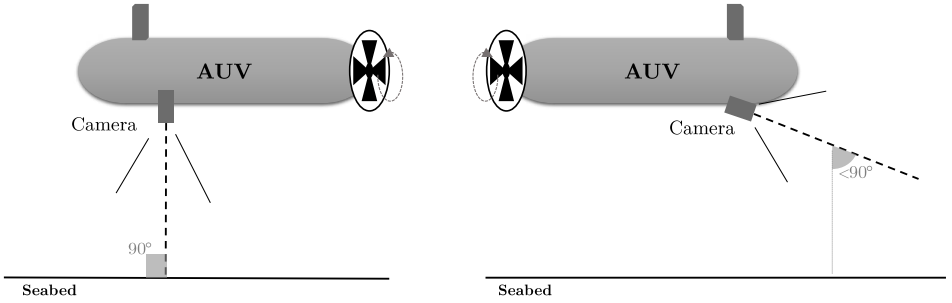


FIGURE 3.15: To handle the non-uniform illumination in underwater environment a variable airlight matrix $\boldsymbol{A}_v$ is used. Each entries of this matrix corresponds to a square patch in the original input image.

m is a new parameter that has to be carefully chosen depending on the scenario. For non-uniform illuminated areas m should be kept low to better handle the light variation. From our test we experimented that values between $\frac{N}{16}$ to $\frac{N}{2}$ (with N the size of the smaller image dimension) are capable to handle the majority of scenarios. In each window the computation of the airlight value is done by taking the biggest intensities on

input image and corresponding to the pixels that are brightest on the map $(1 - s(x))$. In the case that the airlight evaluation window has the same dimension of the input image, we practically return to our initial approach as previously discussed. What is overall important is that the airlight window must be not smaller than the dimension of the patch used to evaluate both scattering and absorption effects. Empirical test suggested us to maintain the airlight window almost ten times bigger.

In underwater environment the illumination is also crucially dependent on depth. Points far away from the camera usually tend to become darker with a certain continuity, so the image illumination will be determined also by the field of view and the angle between the camera and the sea bottom (Figure 3.16). In particular when the principal axis



(a) (b)

FIGURE 3.16: The two opposite camera configurations. When the principal axis of the camera is perpendicular to the seabed (a) the limited depth can lead to evaluate the airlight on larger windows. At the opposite (b), when the principal axis is nearly parallel to the seabed a finer airlight evaluation might be necessary to keep the non-uniform lighting caused by wide depth variations.

of the camera is perpendicular to the seabed—considering an approximate constant depth—few or at least one single window is necessary. Instead, when the input image is characterized by a non-uniform and continuous variation in depth a hight number of windows can better catch the airlight changes.

To conclude the description of our adaptive method it may be useful to point out that considering the airlight as a matrix $(\boldsymbol{A}_v)$, it doesn't change significantly the equation used to recover the image radiance $\boldsymbol{J}(x)$. It is computed as:

$$\boldsymbol{J}(x) = \frac{\boldsymbol{I}(x) - \boldsymbol{A}_v}{\min(t_{low}, \boldsymbol{t}(x))} + \boldsymbol{A}_v \quad . \tag{3.37}$$

To avoid the square-shaped artefacts on the output image (e.g. Figure 3.17) a Gaussian filter is applied before to the matrix A. Figure 3.18 shows a comparison of our method

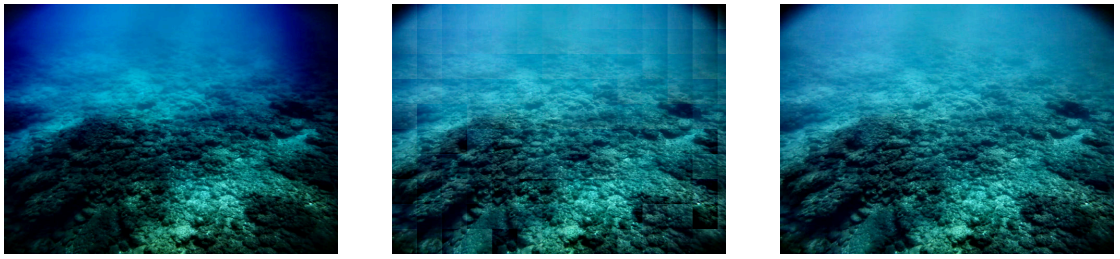


FIGURE 3.17: Example of underwater image recovered. To avoid the square-shaped artifacts (center) on the output image (left) a Gaussian filter is applied to the airlight matrix $\boldsymbol{A}_v$.

with or without using the adaptive airlight estimation.

It is straightforwardly evident as the images recovered with the adaptive airlight estimation present a much more lighting uniformity than the correspondent images, obtained by considering the classical approach that performs only a single sampling of the airlight over the entire image. The native dark areas are instead correctly preserved.

This confirm our initial idea that classical dehazing approaches does not fit well, in general, the underwater environment; not just the transmission estimation function, but also the airlight sampling should be carefully handled.

3.5 Experiments and results

In this section are reported results with our method and are compared with those obtained on the same images by the other major aforementioned approaches: *1)* He et al.'s method, *2)* Drews et al.'s method *3)* Wen et al.'s method. All these techniques have been implemented strictly following the related original works. For this reason we do not take directly in consideration the Carlevaris-Bianco et al.'s approach for dehazing because the lack of sufficient implementation details that might be affect the performance of our version.

In Figure 3.19 are shown some preliminary results obtained with our new method, directly compared with the original image.

From our knowledge there are not widely adopted approaches to quantitative asses the performance of haze removal. Both, terrestrial and underwater techniques are compared by confrontation but there is not a measure that can objectively catch substantial differences. While human observation is capable to determine the quality of a distorted image also without the presence of a reference one, designing algorithms capable to do

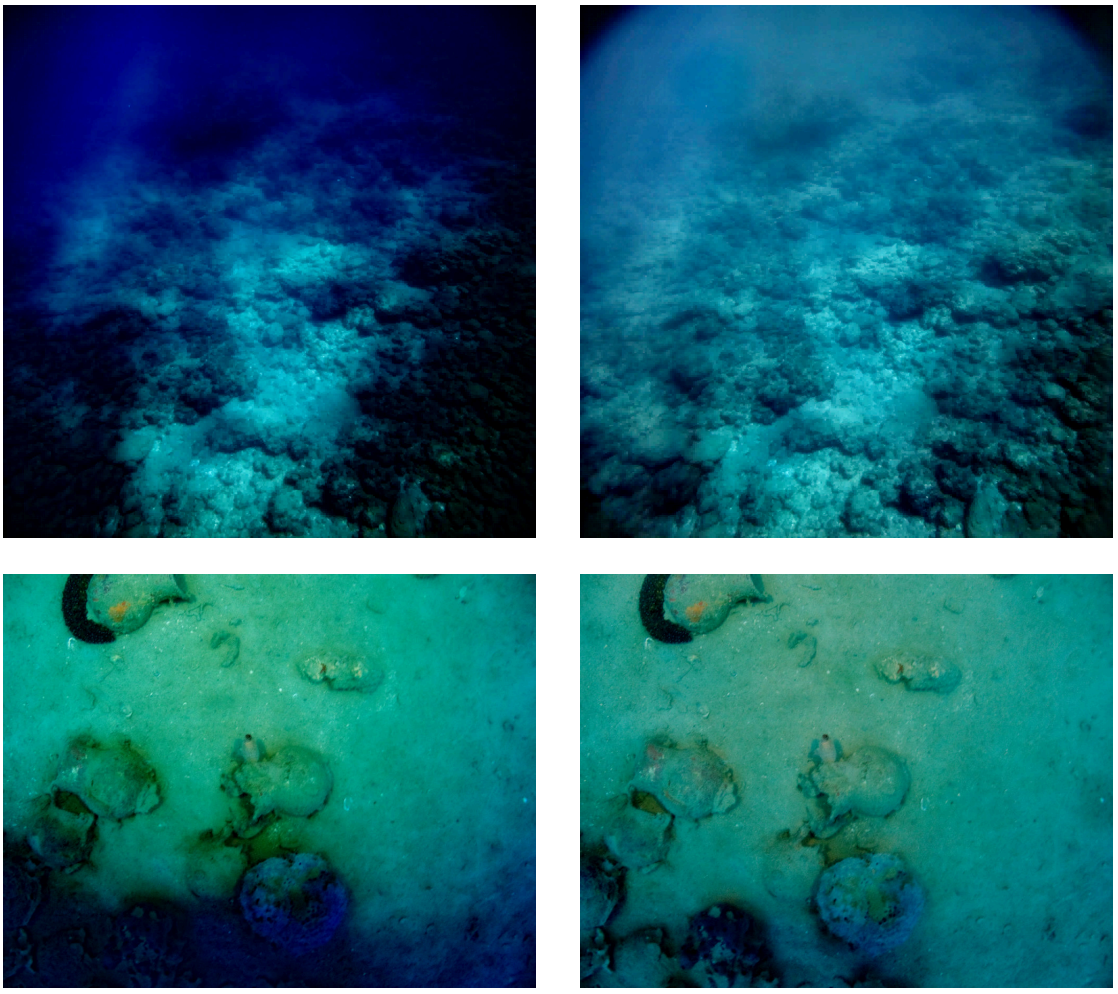


FIGURE 3.18: Examples of images recovered with (left column) or without (right column) using an adaptive airlight estimation. It is possible to observe how, the variable airlight, preserves better the brightness uniformity bot in the case of natural illumination (top row) than in case of artificial illumination (bottom row).

that is still today a difficult task. This is a problem also known as *No-Reference quality assessment*. In some works about the dehazing topic the *Peak Signal-to-Noise* ratio is used even if it may be not a good indicator of overall image quality ([99]), because it is poorly correlated with perceived quality and, mostly of all, it require a reference image that in the case of underwater images is hard to obtain. For terrestrial dehazing are sometimes carried out evaluations based on *ad hoc* setup, simulating the haze in a controlled artificial environment ([100]).

In underwater scenario, some works (e.g. [93]) use known reference pattern to measure the transmission and colour distortion. More than the inherent difficulties to place a reference pattern on the seabed, as we saw in equation 3.13 and also in Chapter 1, the differences in water environment may severe change the behaviour. Drew et al. ([94]) evaluate quantitatively their method by using an image with known ground truth taken

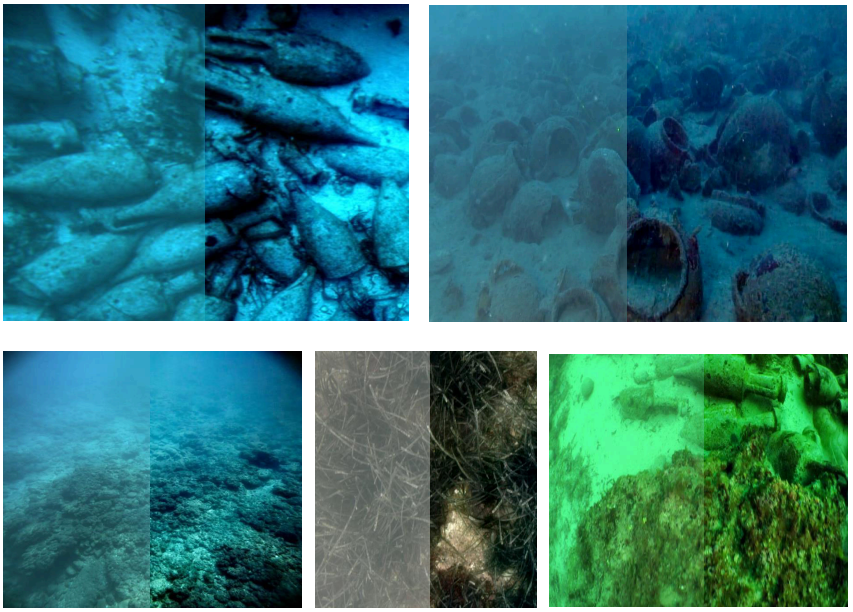


FIGURE 3.19: Other results of the underwater dehazing algorithm.

from *Middlebury dataset*; even if this image do not originate in underwater environment, they simulate on it the water effects. By our part we excluded the option to use an artificial haze over the image because using the same model to add and then remove haze from an image might be trivial and anyhow not much accurate.
Furthermore, given the diversity of approaches for single image dehazing and given the indefinite number of possible scenarios there is no way to proof that a method is actually better than another in every situation.
For all these reasons our approach was to conduct intensive test on our images, directly comparing algorithms over the variety offered by our datasets and using images that are present in other works in literature.
We tested the result of our approach—considering a variable airlight window—related to the three other main algorithms proposed by He, Drews and Wen.
Figure 3.20 and 3.21 collects the obtained results. Each row represents an image taken from one of the dataset described on chapter 4.3 (provided by ARROWS project and/or Soprintendenza Archeologia della Regione Toscana). Starting from the left of Figure 3.20, there are respectively: *a)* Original input image, *b)* He's dehazing, *c)* Drews' dehazing, *d)* Wen's dehazing, *e)* Our proposed method. All results employ an haze evaluation window ($\Omega(x)$) of 15×15 pixels. Our methods computes the airlight on a squared window with dimension a tenth of smallest image side.
The two comparison figures that are shown below were achieved with $t_{low} = 0.2$ and $t_{low} = 0.6$ respectively in Figure 3.20 and Figure 3.21. In fact, after various tests we saw

that the t_{low} is a parameter, shared by all methods, that notably influences the results. It practically handle the amount of the input image appearance that has to be kept in recovered output image; hence for greater t_{low}, input and output image tend to be closer and the radiance is poorly recovered.

As can be noted our method (column *e* on Figure 3.20 and Figure 3.21) seems to perform better in almost all datasets than other approaches. Considering the first Figure with a little lower bound for the transmission (t_{low}), our approach is capable to keep brighter and uniform colours. in low lighting condition, Wen's method (column *d*) notably affects the output colour with high distortions mostly on red channel that sometimes appears overcompensated. Similar effects have been obtained also by changing its input parameters like the patch dimension used for the transmission estimation. On the other hand, the Wen et al.'s is the approach that more than the others mitigates the blue (or green) colour predominance in underwater images.
He et al.'s method shows consistent colours, however its performance is lower than in terrestrial environment.
Our method instead has better overall performance, but in particular the differences are much more remarkable in *D3*, *D12* and *D11*. The positive effect in using a variable airlight estimation seems to be confirmed by the good results obtained with darker images as for example in dataset *D12*. All images are characterized less or more by a more lighting in passing from $t_{low} = 0.2$ to $t_{low} = 0.6$. Drews et al.'s and Wen et al.'s methods are those who seem suffer more this variation, especially with input image having lower illumination as images in datasets *D2*, *D3*, *D6*, *D11* and *D12*. By increasing t_{low}, the Drews's and He's methods obtain closer output results in practically all datasets as it is possible to observe in Figure 3.21.

Figure 3.22 shows the differences that we achieved by using our approach with a variable (column *b* and *c*) or fixed airlight (column *a*). While there are few or less differences between the three approaches when the image has an uniform illumination (e.g dataset *D6* and *D9*) using a variable airlight estimation allows to get better results when the depth variation in the input image is higher, as for example in dataset *D3* and *D12*. In particular, images on the right column in Figure 3.22 globally have a better uniform lighting in comparison of those that use a single airlight sample calculated over the entire image. Images on the central column have a larger airlight windows (2.5 times bigger) and their illumination is slightly lower than using a finer airlight sampling. By comparing these images, obtained with $t_{low} = 0.2$, with the corresponding ones in Figure 3.20, we may observe that our proposed method, also without the adaptive airlight windows, is able to perform better in underwater environment than the Drews's and Wen's approaches especially in dealing with images characterized by larger depth ranges. Obtained results, instead, are much more close when the image depth is limited.

All our implemented algorithm for single image dehazing are in Matlab® code. To process an image of about 3 megapixels, it takes about 200 seconds, comprehensive of the airlight estimation and refinement steps. This time can be further reduced by employing the "fast" version of guided filter ([96]) to refine the transmission. With a limited number of windows for sampling the airlight, the temporal difference between the twice version of our algorithm is negligible.

From all the reported comparisons—and the many others that we conducted—over multiple images taken from our datasets we observe that our proposed method, especially with variable airlight estimation is capable to obtain in general better results than the competing approaches in underwater environment.

As already said, speaking about the problem of underwater dehazing and its peculiarities, unlike the terrestrial scenario, for this environment there is a lack of datasets that can be employed to actually asses the overall performance of existing methods.

Literature works, including those we analysed and implemented here, lack of a common shared evaluation dataset publicly available.

In figure 3.23 are presented some results with images taken from the works of Wen ([6]) and Fattal ([5]). Here are compared results obtained with our proposed algorithm (column D), Carlevaris-Bianco's (column B) and Wen's (column C) dehazing approach. In the image on top our method seems to perform close to Wen's one. On the other hand, the bottom image points out as our method has apparent lower performance on the image background. In particular, differently from other methods, it does not substantially change the image colours. We may notice that other methods (on columns B and C) recover colours in a quite distorted and not realistic way, while our approach is able to keep more image consistency. In Bianco's method, for example, colours tend to be whiter while in Wen's work there are some appreciable distortions due to the lighting variation.

Moreover there are some reason behind these differences. Except for our output image (column D), all the others are taken from ([6]). Authors do not specify information regarding the input image used for these results and our starting image was available only in a very compressed and small (300×240 px) form. In relation to the haze phenomenon and its diffusion/distribution across an image, compressions (lossy) or large resizing may severe infer the performance of dehazing algorithms. It's a fact that our method works better with unfiltered and uncompressed input underwater images. Furthermore, in Figure 3.23, there are no sufficient information about used parameters, optional colour enhancement or special pre-processing filters. In dealing with terrestrial images, for example, *white balancing* algorithms are often used preliminary for the dehazing algorithm; in our approach we don't use anyhow extra enhancement or filter because we observed

that depending on the particular environment they can lead sometimes to evident mistakes and may hide the actual ability of our proposed dehazing algorithm.
The original datasets that we used and presented here (compatibly with copyright permissions because the majority of our images come from underwater videos taken during the *ARROWS project* funded by the European Commission through the *7th Framework Programme for Research and Technological Development*) jointly with our algorithm, will be publicly released to give a shared dataset for evaluation and to improve further works on underwater dehazing topic.

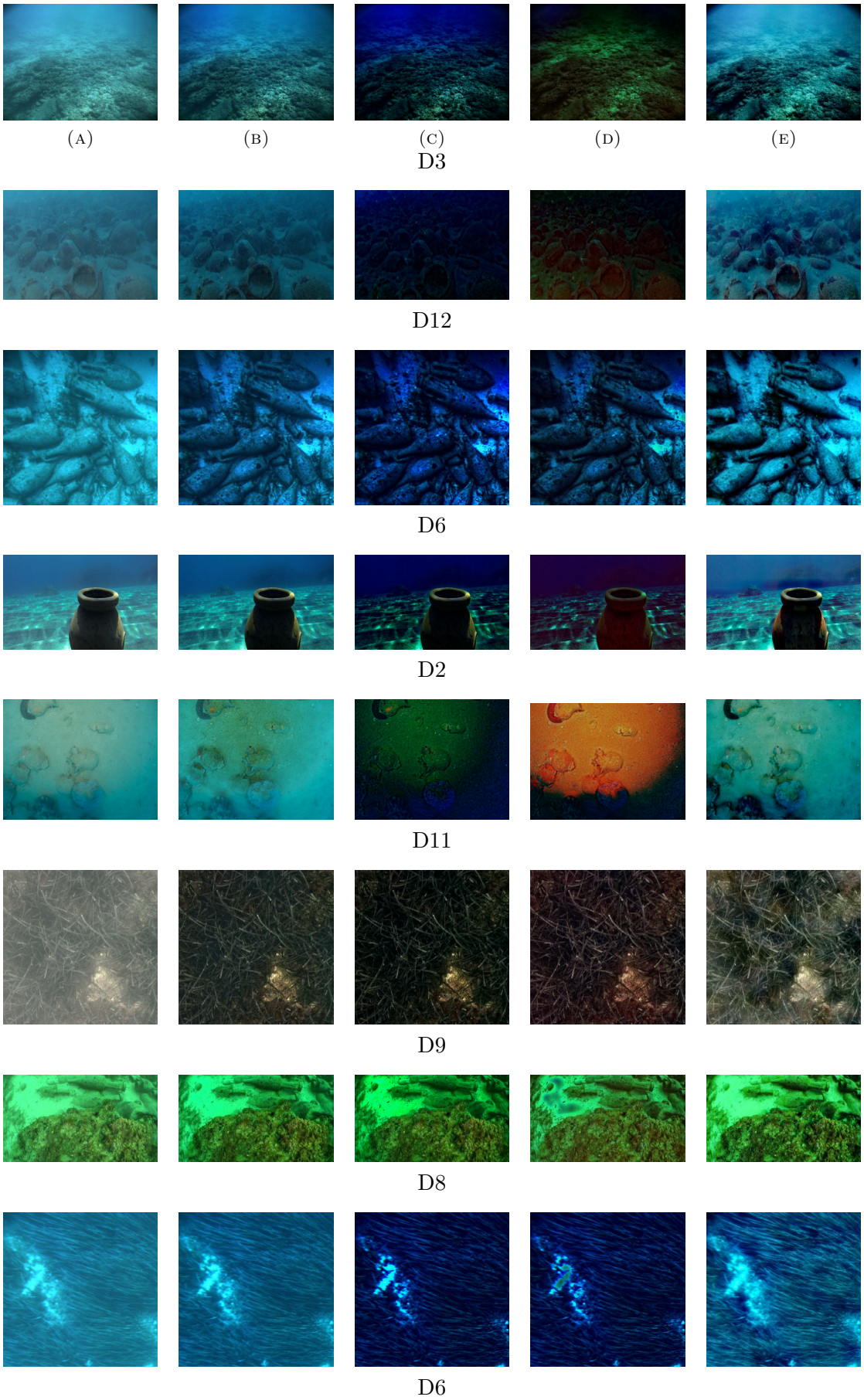


FIGURE 3.20: Final results obtained for input images (column *a*) applying methods: *b)He*, *c)Drews*, *d)Wen*, *e)Our*. Images are recovered with $t_{low} = 0.2$, $\Omega(x) = 15$.

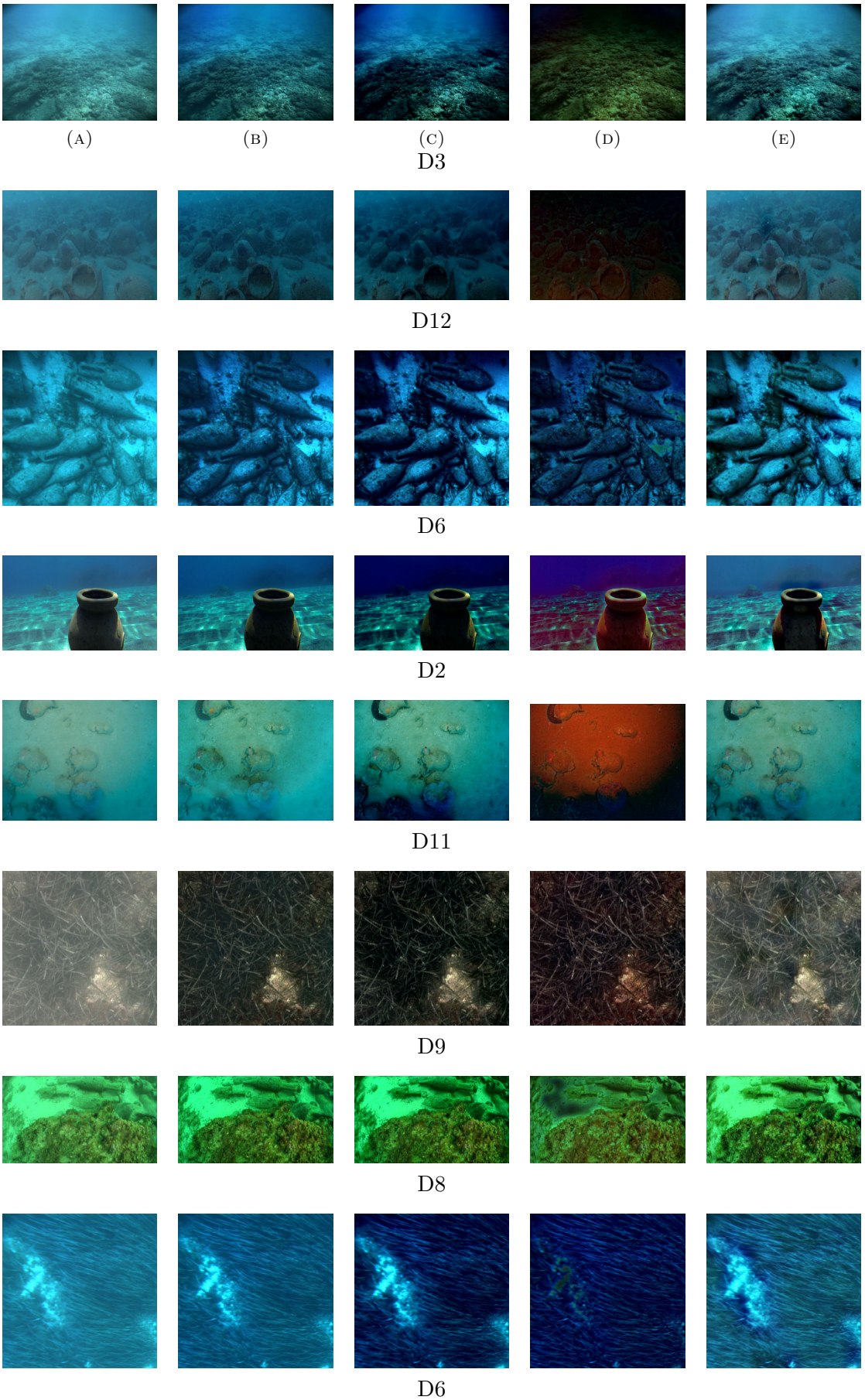


FIGURE 3.21: Results with $t_{low} = 0.6$.

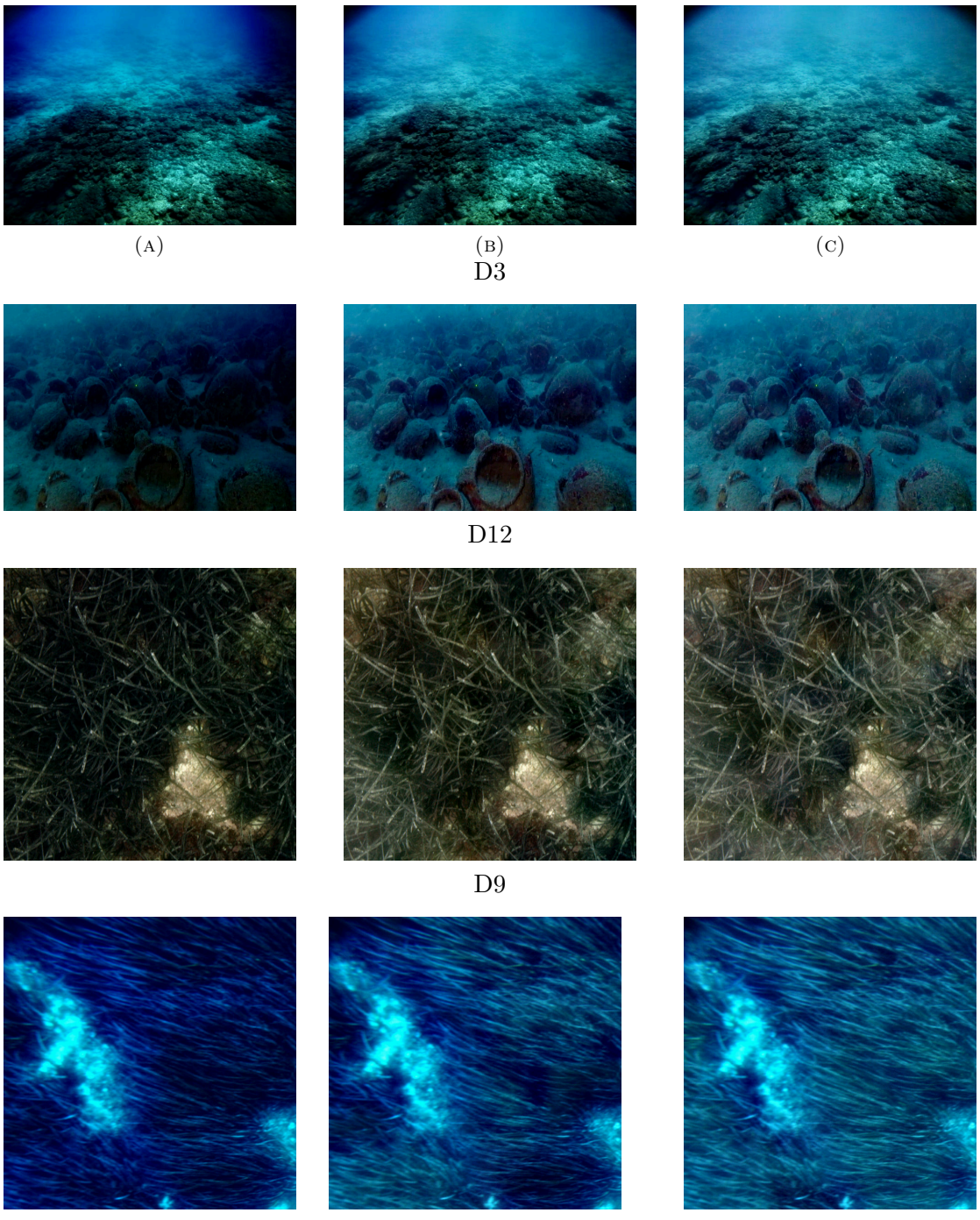
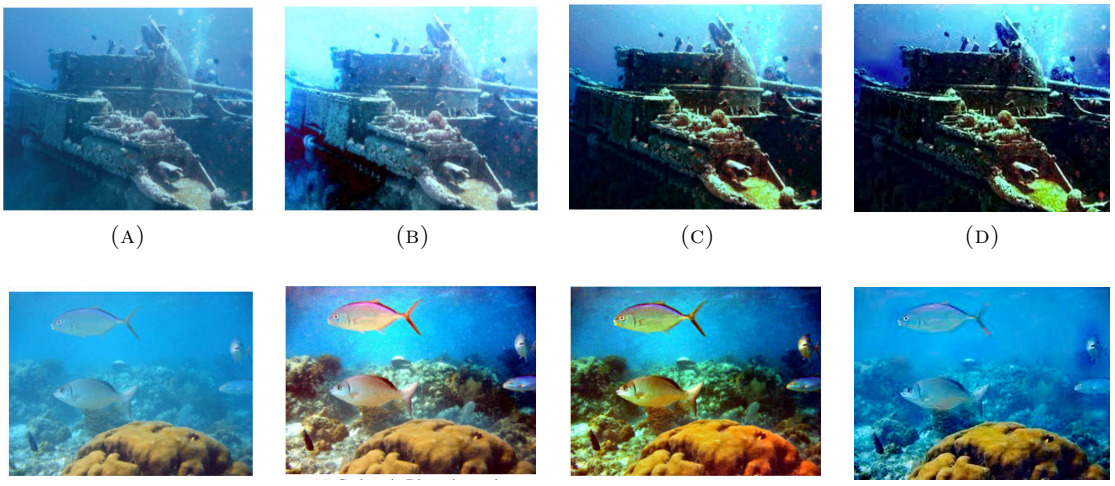


FIGURE 3.22: Comparison of our method using a variable (column b and c) or fixed (column a) airlight estimation. Main parameters: $t_{low} = 0.2$, $\Omega(x) = 15$, and the airlight window is a quarter of the smallest side image (column b) or a tenth (column c). While there are less differences between images in presence of a uniform illumination (dataset *D6* and *D9*) using the variable airlight allows to asses better results when the depth variation in the input image is higher (*D3* and *D12*). We can also notice how reducing the airlight window size the output image acquires a more uniform illumination.



(A) (B) (C) (D)

FIGURE 3.23: Example of comparison of our method (column D) with Carlevaris-Bianco's (column B) and Wen's (column C) dehazing approach, in relation to the input image (column A). Images are taken directly from the work in [6]. In the first row our method (with $t_{low} = 0.2$ and approximatively 40 airlight samples) perform close to the Wen's method. In the second figure (bottom row), our approaches works well on the foreground but affect only slightly the background, but differently from other approaches, colours are not highly distorted.

3.6 Dehazing for coarse depth estimation

As we said at the beginning of this chapter, the haze removal is intrinsically linked to depth information of a 3D scene. In particular, in equation 3.1 we saw a close relation between depth and image appearance in hazy environments.
Haze or fog, when is not occluding entirely a scene can be seen as cue for recovering coarsely the scene depth [101].
There are various method to obtain a depth estimation from a single image, *shape from texture*[102], *shape from shading*[103] or by assumptions on the presence of particular known object or structure inside the scene ([104]). In [105] the problem of inferring the depth of an image starting from a single image is carried out in a more general way by a supervised learning approach. Starting from a number of training examples with a known ground-truth they learn a classifier (see also [106]) based on the analysis of a set of visual clues.
The haze approach is today not enough well suited in image processing ([86]). Being able to recover—with some limitations—the actual scene radiance from an hazy image, we also need to carry out some information about depth. Theoretically this is a relative depth because unless a more precise model and absolute quantifications this method is able to catch only a coarse depth map of the environment. Some errors might be pointed out also by particular image configuration so, for example the sky, snow, sea or rivers reflections can lead to incorrect estimates.
Practically, every image taken in a terrestrial environment is affected by haze. Only a vacuum atmosphere (i.e. the absence of a medium between the target and observer) is theoretically immune from it. Here shapes lying on different-depth planes might be caught by the transmission map. Figures 3.24, 3.25 and 3.26 show results obtained with the He et al. method for terrestrial images. The presence of a smoothly distributed fog as on Figure 3.24 allows better results than in case of a low hazy atmosphere as in successive Figures (3.25 and 3.26). Anyhow the foreground scene is well segmented and the results are comparable to those obtained with more articulated approaches. Clearly the haze-based approach works better with outdoor images, but it is possible to extract acceptable results also for indoor scene as reported in Figure 3.27. We used an image (the top one) taken from the Middlebury Stereo Dataset ([7],[8]), and we compared the results that we obtain with our implementation of the He et al. method (images on column right) in relation to the effective depth map (left column). Clearly there are some errors, mostly due to the high-textured background and the small depth range of the image, but the foreground object is still appreciable.
Switching the scenario from terrestrial to underwater it is possible to apply our method to recover a coarse 3D map of the seabed. Underwater images are in general affected by

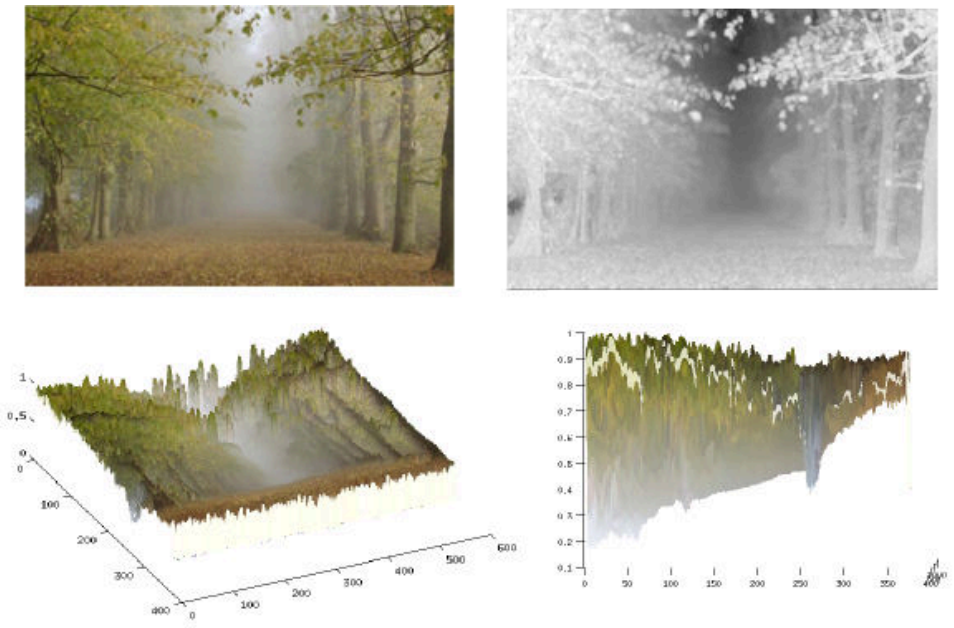


FIGURE 3.24: The transmission map can be seen as a cue for estimate the (relative) depth, from a single foggy image.The top right image is the computed transmission for the input image (left). Whiter points represent closest point. Below is reported the obtained texture-mapped surface seen by two views. (Image from [4])

much more distortions and the visibility is typically lower and all this might affect the final result. Differently from the terrestrial method for haze removal (where only the scattering effect is considered) in underwater images we employed three transmission maps (one for channel) and consequently we have not a single transmission leading straightforwardly to a depth estimation. To overcome this limitation and to obtain a coarse 3D depth map of the underwater environment the idea is to use a transmission map derived by taking the minimum among all the channels. In particular for every point x of the transmission $\boldsymbol{t}(x)$, is computed:

$$t_{depth}(x) = \min(t^{red}(x), t^{green}(x), t^{blue}(x)) \quad . \tag{3.38}$$

The following images (from 3.28 to 3.35) report the obtained depth-transmission map (central image) obtained for several significant underwater scenarios (left image). All images are processed using our method with estimated haze over a 15×15 pixel window. The corresponding texture mapped on the depth surface, derives from the original (non-dehazed) image so that the resulting appearance is not influenced by the choice made for the other parameters (like t_{low} or the airlight window size). Considering that low (darker) transmission values are representative of many further 3D points, we obtain consistent results both considering large (e.g. Figure 3.29) than small (e.g. Figure 3.31) depth variation ranges. We often use the adjective *coarse* to underline that this

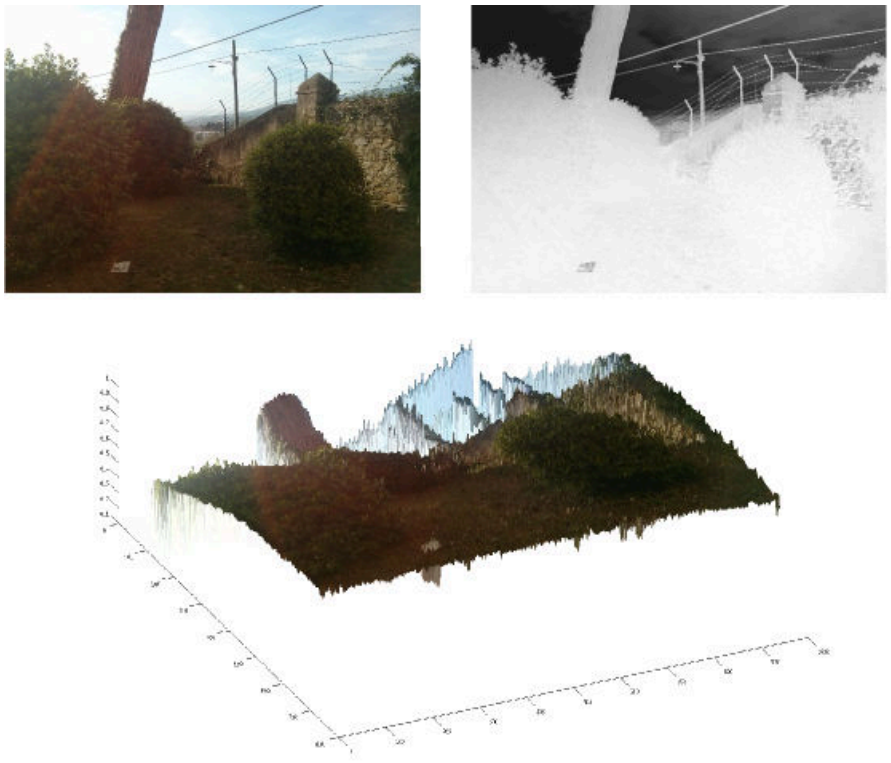


FIGURE 3.25: A second example of a texture-mapped surface obtained starting from a single image. Without any further information using the haze distribution this method is able to segment out the scene foreground, also in presence of a moderate haze.

recovered depth is not composed by precisely trusted values. Furthermore, the evaluated measures are not so accurate to give the real depth measure, both absolute and relative. What this approach can do, instead, is to provide, a background/foreground segmentation and a sketch of the scene structure.
For what concerns the errors carried out by this approach we observed that as in figure 3.32, 3.34 and 3.35, lighting saturated area may lead to inconsistencies, in most cases limited to spotted zone of brighter transmission. Also the artificial light, in stronger area, might lead to bad estimations as for central objects in figure 3.30. Note that the first two figures (Figure 3.28 and Figure 3.29) are characterized by an apparent distortion in correspondence to the image corners. Actually the transmission is here correctly evaluated because the four visible dark corners are due to the circular camera housing mounted to the AUV, and the position of those points is really closer to the camera. The smoothly curved surface is then the result of the application of a filter that limit abrupt transmission changes.

Despite its coarseness and limitation, haze removal can bee seen as a fast, simple and effective way to compute a coarse relative scene depth just starting from a single

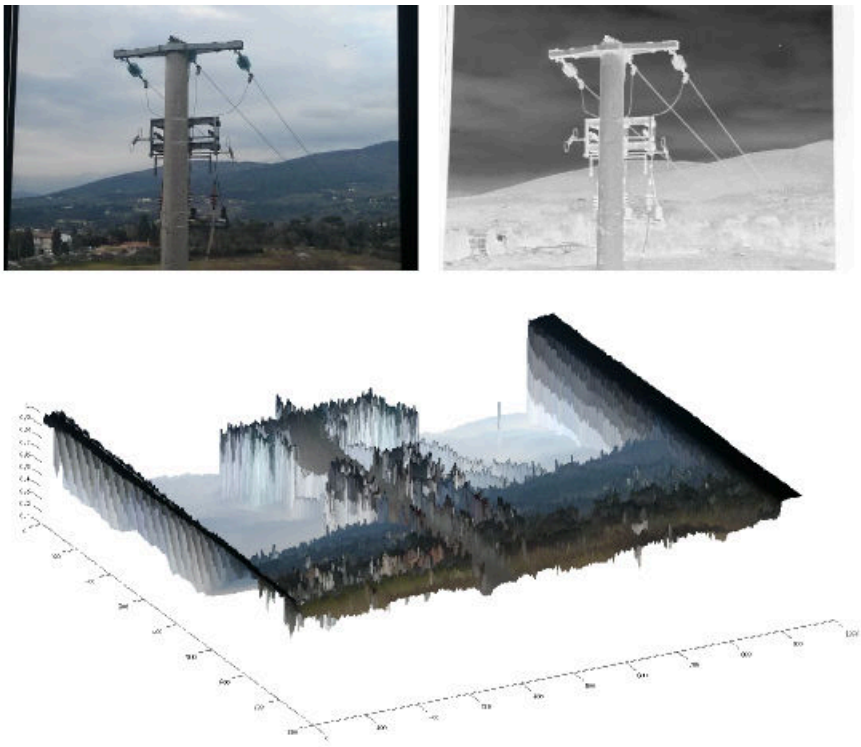


FIGURE 3.26: Another example of coarse depth estimation using the He et al. method.

natural image, without particular informations needed or special parameter tuning. Hypothetical applications may be in all those visual patterns where there is the need to get an overall scene sketch in a limited time, for boosting already existing algorithm or simply where there is not enough information (i.e. images) to run more sophisticated method for obtain depth informations.

The use of the haze effect like a cue for coarse depth estimation does not ends here and other applications can be derived. Another example is employing it in the image forensic field.

As many natural phenomena captured by images (e.g. lighting condition in [107]), the process of correctly and consistently replicate haze distribution is not easy to achieve and for this reason it is reasonable to think haze as a clue for detecting potential inconsistencies over an image.

In particular some digital forgeries may be discovered in where an image presents added elements that they were not in the original one.

Starting from a single tampered image and taking its transmission map the relative mutual position of every object that is present into the scene is considered. The position of these elements should be coherent with the perspective (i.e. the actual position) in the original image and every inconsistency may be regarded as a potential forgery. Figure 3.36 shows a tampering action that can be detected, by looking at the inconsistencies

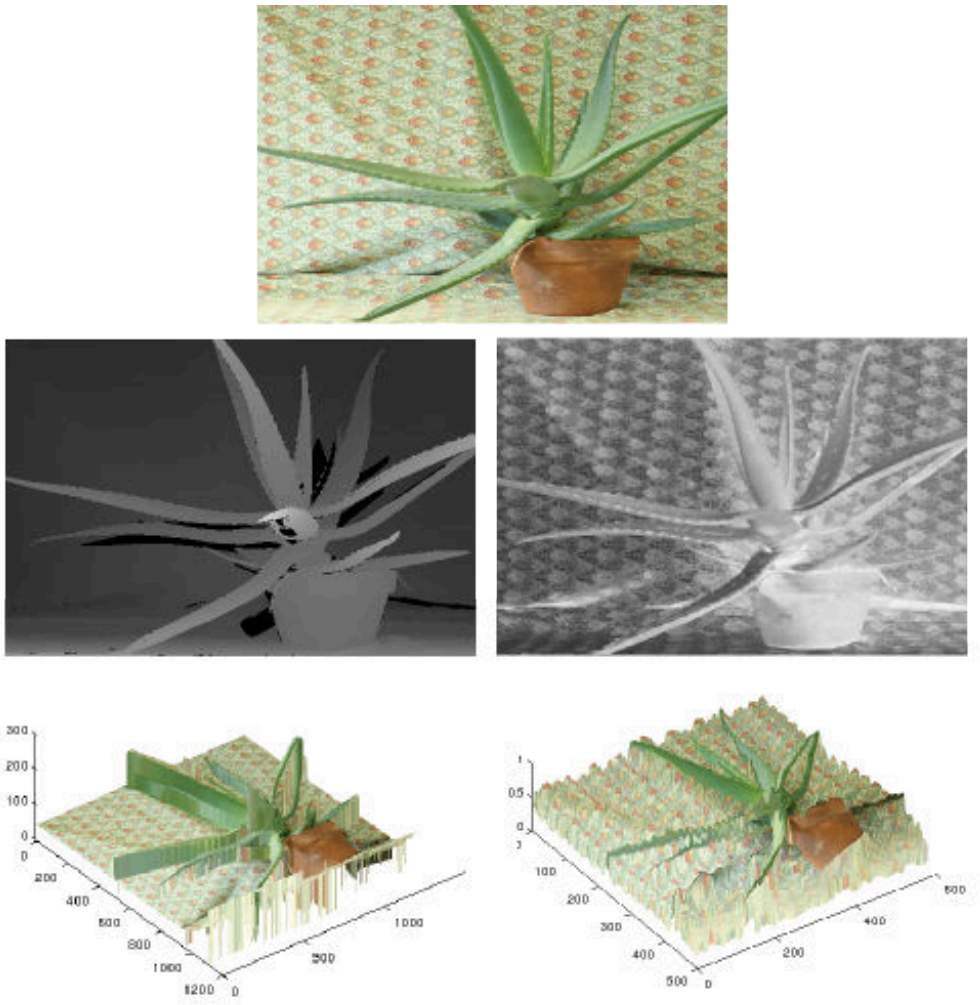


FIGURE 3.27: Example of coarse depth estimation for an indoor image. The (top) input image is taken from the Middlebury Stereo Dataset ([7],[8]) then we compared the results that we obtain with our implementation of the He et al.'s algorithm (images on right) in relation to the effective depth map (left images). Although there are some errors due to the highly textured background the foreground object is appreciable.

generated by the transmission map and the actual perspective.

Clearly this approach—as many others in image forensics—needs the human intervention to label the original image with the real relative mutual positions of objects.

This approach of using haze as a cue to discover potential image forgeries is based on the fact that elements extracted from different images, taken in different context and time would be in general affected by distinct haze distributions. Hiding such modifications is not an easy task to accomplish and it may represent a weakness leading to discover whereas an image tampering action has been occurred.

FIGURE 3.28: Example of coarse depth recovering from a single image. Figure on the left is the input image, the central one represents the transmission map and the right is the textured obtained surface. The fact that all the four image corners appear ahead the other image points is due to the circular camera housing employed that partially occludes image corners. So this is not properly a distortion but instead those dark four image corner are correctly placed; anyhow some distortion is inducted by the filtering step that smooths the abrupt depth changes as in this case.

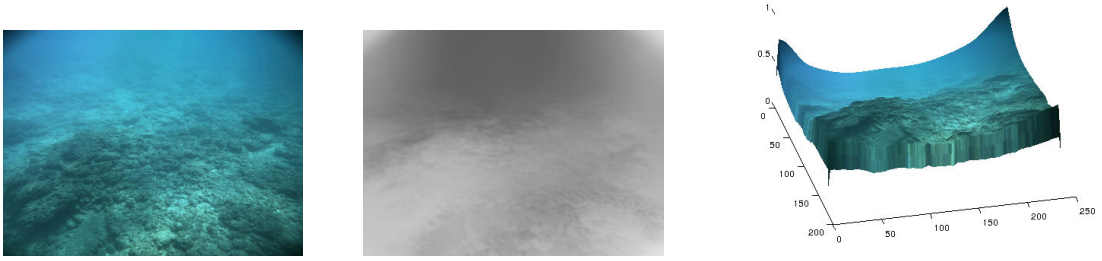


FIGURE 3.29: A second example of underwater scenario with natural, non-uniform illumination.

FIGURE 3.30: Example of 3D coarse depth recovery in a scenario with artificial illumination. We notice that a strong illumination may deceive the foreground/background estimation as in the case of the object in the center of the (left) input image whereof presence is not correctly captured and appears almost fuse with the seabed.

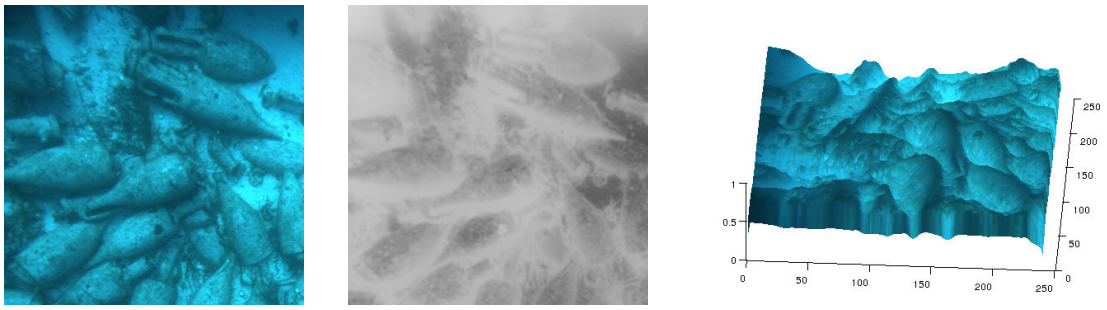


FIGURE 3.31: An example of 3D coarse scene depth using an image with a small depth range variation. Even if some evaluation errors are present (mostly due to the object surface reflection properties and the strong blue presence), the overall depth evaluation is quite consistent and realistic.

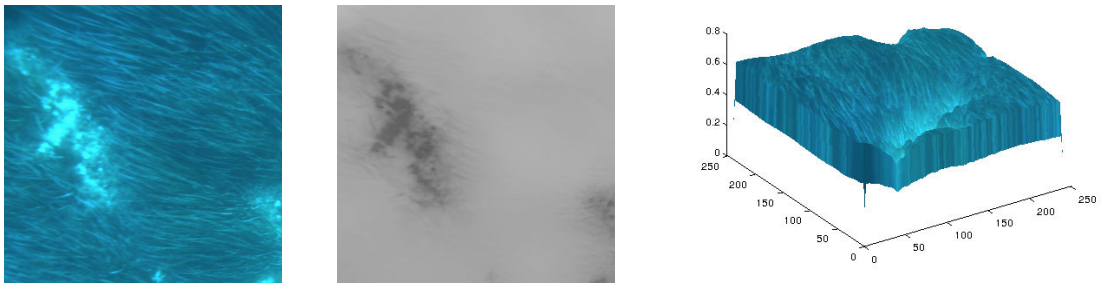


FIGURE 3.32: Another example of depth evaluation. This is a detail of a bigger image again strongly dominated by blue channel.

FIGURE 3.33: Example of depth evaluation. In this case the image is dominated by the green colour, but result is comparable with the previous obtained.

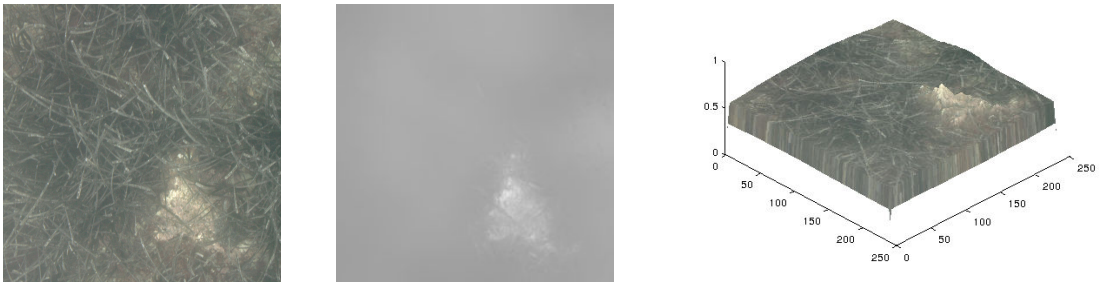


FIGURE 3.34: Another example of depth estimation with a naturally coloured image.

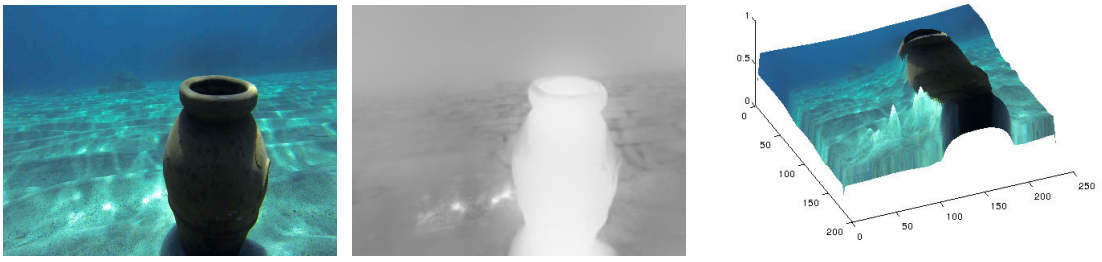


FIGURE 3.35: Example of coarse depth estimation in an image characterized by a clear foreground object. We can notice (both on central than right figure) that the foreground object is correctly identified by the transmission map. As happened in other images, some illumination saturated areas, like those near the central object, might appear as foreground even if they are not.

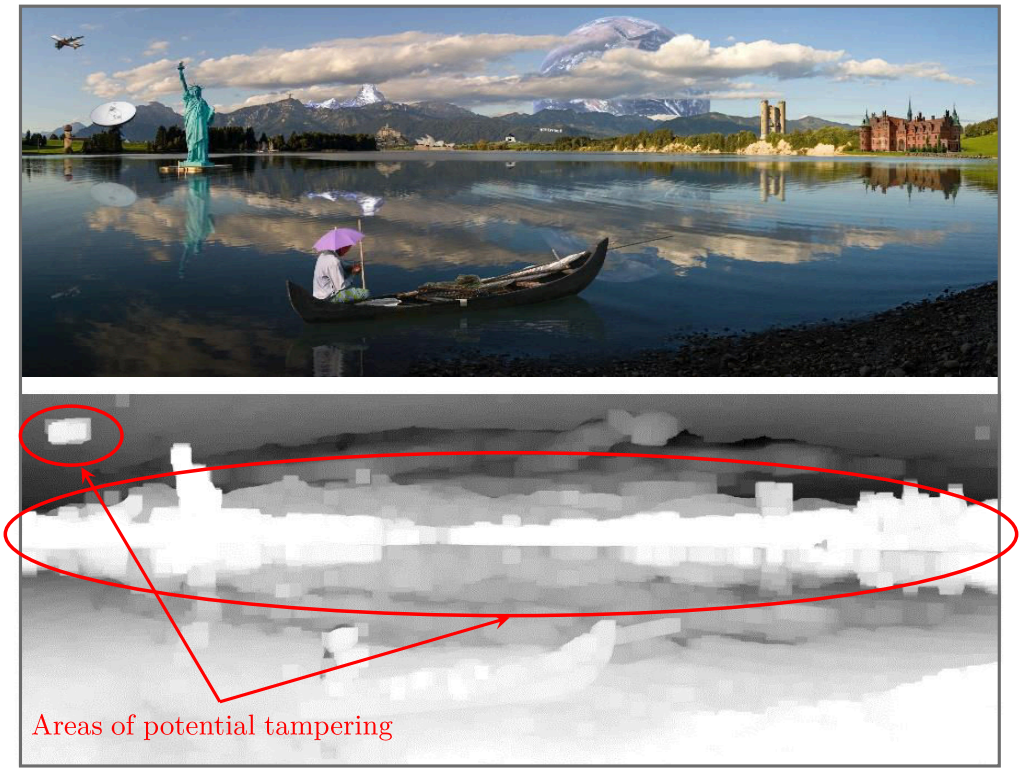


FIGURE 3.36: Example of potential tampering (the original tampered image is from Wikipedia).

Chapter 4

Underwater classification: algorithm and feature comparisons

Texture analysis is an active research topic in the fields of computer vision and pattern recognition. There is not an unique and widely accepted definition for texture, even if related studies were conducted from the beginning of image processing. Many methods characterize textures regarding features extracted from them.

First of all, texture analysis is useful to describe images that shows regular pattern. Natural environment shows a lot of coarse and irregular shapes at different scale and resolution (Figure 4.1). On the other hand, places in where humans live are subject to transformations that make the visually perceived environment characterized by well defined, identically repeated, finer and regular shapes. We can hence define human world as characterized by *object-oriented* scenes in opposition to the more *pattern-oriented* scenarios of the natural world. Therefore if we want to deal with this latter, we need a representational framework that is able to describe effectively such shapes.

Underwater environment is still mostly unexplored and so it is a place where nature can totally express itself. Performing task as segmentation or classification in the underwater scenario can be very difficult because imaging conditions as illumination and magnification and different noise sources contribute to a poor image quality.

FIGURE 4.1: Examples of natural patterns (Images from Flickr).

4.1 Related works and background

Many times researchers have attempted to emulate the ability of the human brain to understand the content of images for the interpretation of images and developing image understanding algorithms for applications including robotic vision, remote sensing, assisted medical diagnosis and automated target recognition. Texture analysis is strictly connected to human low-level perception tasks. Historically from its beginning, image analysis took into consideration to deal with textures.

It is important to notice that the term *texture* is not a prerogative of the visual image analysis—meaning the images that are acquired through CCD (Charge couple devices) or CMOS (Complementary Metal-Oxide-Semiconductor) sensors—because many techniques are suited also in more general frameworks of signal processing. There is not a commonly adopted definition for what a texture is [108]. Simply we can define it as the main and direct source of visual information. Texture are complex visual patterns made itself by entities or sub-patterns in a hierarchical fashion. What characterize a texture are apparent properties regarding density, uniformity, roughness, smoothness, directionality, frequency, phase, etc.. . It is possible to say that an image region has a constant texture if a set of given local properties in that region is constant and is slowly varying. It must be underlined that the texture has both local and global meaning. The analysis typically require the identification of proper attributes (features) that can differentiate textures during classification, segmentation and recognition tasks.

Selecting textural feature which are independent and discriminable may aid an eventual preliminary segmentation process. Textures present significant properties in digital imaging system and have an important role in human visual perception as also experimented in [109] from a psychological point of view and where textures are addressed as

an important cue for scene identification.

Texture plays an important role for the analysis of many types of images. A variety of measures for discriminating textures have been proposed. Most of them quantify the texture measures by using just few values; all the extracted elements are then combined as elements of feature vectors for performing classification or discrimination tasks and with or without knowledge about the imaging conditions ([110]).

In [111] is proposed an algorithm that uses textures jointly to colour descriptors over multiple scales to perform an automatic annotation of underwater images taken in coral reef scenarios. Althougth some limitation the final aim is here quite similar to the one proposed in this work, with using SVM to accomplish the proper classification task.

In a classification framework the texture and local spatial statistics are actually an important way to describe the information that is present in an image, as evidenced in [112]. The main conclusion is that both texture and local spatial statistics are able to improve the classification accuracy. Other than by features this latter is influenced also depending on the chosen window size, resolution and direction. Here the application field is not the underwater environment but the satellite imaging that anyhow shares a lot of similar properties and characteristics with it.

Already in [113] a deep investigation about the GLCM was presented that by considering their statistical meaning showed as only a small subset of them, in common application, must be regarded as effectively needed.

Wavelets are an alternative approach widely adopted for texture recognition and or classification, as shown in [114], where three different approaches are analysed. Similarly in [115] the *Haar wavelet transforms* are used in combination with Support Vector Machine classification or in [116] where the wavelet technique is directly integrated in the SVM kernel function.

In [117] a statistical versus wavelet-based approach was instead carried out. The point is that even if features based on first and second order statistics are characterized by far less number of components in comparison of methods based on wavelet transforms, the statistical features are able to show better performance. Statistical features does not also involve computational intensive transformation so are in general suggested in time constrained tasks. The experiments have been conducted on medical images but can be extended also to other similar non-regular shapes.

In [118] the *TextonBoost* approach is presented. It assumes a discriminative model for an efficient and effective semantic segmentation and/or recognition of images. A similar discriminative method, specific for underwater images has been also presented in [119] for detection and image segmentation. Even the good achieved performance, both approaches are anyhow too object oriented to be effectively employed for a seabed classification.

Ideally every 2D image patch can be seen as a texture, but in general only those that present some regularities are suitable in texture analysis. Seen as an informative signal, each 8-bit grey-level image patch with $N \times M$ pixel can represent $256^{N \times M}$ different patches. Without considering problems due to viewpoint changes, images taken from the real world are affected by a strong intrinsic variability, in acquisition process and atmospheric-induced noise. We can isolate two major relevant problem regarding texture analysis: *Discrimination* and *Classification*. The first one (see [120] and [121] for good reviews) is ascribable to a segmentation problem.

A texture is not a bounded entity but it is a region defined by some of its characteristics and their homogeneity. So, how can we discriminate textures? The answer is strictly related to the actual task that we need to accomplish. It may depend from different factors like desired resolution or invariance properties in relation to image transformations. Method based on *quad tree* [122], *Markov Random Fields* [123] or *Superpixel* [124] are some of most famous for image and vision tasks.

The classification problem is meant to grouping textures about some desired features. The difference with the discrimination is that there is a fixed set of texture classes among which choose. To accomplish tasks as texture description, discrimination, classification and retrieval there is the need to compress texture information in a (relative) small subset of simple, robust, reliable and well defined measures. For this reason texture statistics are largely employed.

According to [125], [126] and [127] there are various method for approaching texture analysis. Basically there are three categories: 1) *Structural*, 2) *Statistical* and 3) *Model based*.

The *structural* methods [128] try to represent a texture with some basic primitives and their spatial placement. Examples of basic primitives can be the pixel tone or edge direction. During time various methods have been developed and in addition, spatial relations may be considered to improve the discrimination factor. *Statistical methods*, as the name said, define texture only by a set of local statistics based on pixel gray-levels with a single channel at time.

These approaches are sometimes related to the structural one when statistics are used to defines primitives without attempting to describe explicitly the texture. *Model based* methods are instead directed to describe an image texture with a linear combination of basis functions or with probability models chosen *a priori*. The texture is in this case defined by the parameters or coefficients of these models. Examples in this category are methods based on *Simultaneous autoregressive models* [129], *Markov model* [130], *wavelet based* [131], *fractal dimension* [132], and *Gabor filter* [133].

4.2 Classification and texture analysis in underwater

As previously said textural analysis is a early and wide field of study in image processing and pattern recognition. Obviously each method has its strengths and weaknesses considering the numerous way in which features are extracted and used.
There is not a straight way that was proven better than others for an effective use in general texture-based tasks of segmentation/classification. By that, in this work, we are interested to investigate the behaviour of some textural features to handle with the underwater environment that has the properties already described in Chapter 1.
In the work proposed in [134] the authors aims to classify seabed using textural features. With some example comparing the *co-occurrence matrices (COM)* [135] with *self organizing maps (SOM)* [136] they found that even if SOM features are lighter and easy to compute the COM-based (*contrast*,*correlation*,*Angular second moment* and *Inverse difference moment* appear to better handle with the underwater noise.
A deep investigation of performance of all proposed methods during years is practically intractable, so based on some main works about underwater classification we have been focused only in a subset of them.

4.2.1 First order statistics

Given a single-channel $N \times M$ image it is possible to compute the corresponding histogram as:

$$h(i) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \delta(f(x,y), i) \tag{4.1}$$

where

$$\delta(i,j) = \begin{cases} 0 & if \quad i \neq j \\ 1 & if \quad i = j \end{cases} \tag{4.2}$$

is the *Kroneker's delta* with $i \in [0, i_{max}]$ and (x, y) that represent the point coordinate. Dividing by the product $N \times M$ the equation 4.1 it results,

$$p(i) = \frac{1}{NM} h(i) \tag{4.3}$$

where $p(i)$ may be seen as an approximation of the probability density function. Using only the histogram itself to characterize textures on real images is not a viable choice. It lacks of stability in relation to little variations and noises that might affect the image

acquisition process. For this reason the histogram is further described by some statistical measures extracted from it. These values are employed as textural features, and are more qualitative than using the entire histogram itself. In image processing literature they are often referred as *first order statistics*. Considering the usual 8-bit image representation $i \in [0, 255]$, the common statistical measures extracted from an histogram are:

Mean:

$$\mu = \sum_{i=0}^{255} ip(i) \tag{4.4}$$

Variance:

$$\sigma^2 = \sum_{i=0}^{255} (i - \mu)^2 p(i) \tag{4.5}$$

Skewness:

$$\mu_3 = \sigma^{-3} \sum_{i=0}^{255} (i - \mu)^3 p(i) \tag{4.6}$$

The *skewness* is a measure of symmetry around the histogram mean value μ. It is zero when there is a perfect correspondence.

Kurtosis:

$$\mu_4 = \sigma^{-4} \sum_{i=0}^{255} (i - \mu)^4 p(i) - 3 \tag{4.7}$$

As the *skewness*, the *kurtosis* index describes the shape of the distribution. It indicates the *flatness* or *peakedness* property of the histogram with respect to a normal (Gaussian) distribution. In statistical theory the definition is not unique and there may be multiple different ways to define it. In this representation the proper Gaussian distribution has a kurtosis equal to zero.

Energy:

$$E = \sum_{i=0}^{255} p(i)^2 \tag{4.8}$$

Entropy:

$$H = \sum_{i=1}^{255} p(i) \log_2 p(i) \tag{4.9}$$

The *entropy* may be regarded as a measure of the distribution uniformity.
In addition, other but anyhow less relevant features, used to characterize images at the histogram level are *Median*, *Maximum* and *Minimum*.
For some measures, may be a good choice to standardize the image values first in order to have $\mu = 0$ and $\sigma^2 = 1$. In this way the successive comparisons might be more effective. Note that an histogram normalization is also required when the compared images have not the same dimension.

4.2.2 Second order statistics

In the image processing field the *second order statistics* are those measures that derive from a joint probability distribution of pairs of pixels. Differently from the previous, the second-order statistics, also rely on the spatial positioning of pixels.
Starting point is the definition of the *co-occurrence matrix* (or *Gray Level Co-occurrence Matrix*, GLCM) that is indicated for single channel image $I(x, y)$ with values in $[0, 255]$ as:

$$h_{d,\theta}(i, j) = \sum_{x=0}^{255} \sum_{y=0}^{255} g_{d,\theta}(x, y) \tag{4.10}$$

where,

$$g_{d,\theta}(x, y) = \begin{cases} 1 & if \quad I(x, y) = i \quad and \quad I(x + d_x, y + d_y) = j \\ 0 & otherwise \quad . \end{cases} \tag{4.11}$$

In this latter equation, $d = (d_x, d_y)$ is the *distance* between a pair of pixel and θ is the *direction* (i.e. angle) in which this distance is measured. In this way d can be formally expressed as:

$$\begin{cases} d_x = d \cos\theta \\ d_y = d \sin\theta \quad . \end{cases} \tag{4.12}$$

The co-occurrence matrix is inherently square. Note that the parameters d and θ lead to a very large number of different matrices. As we'll see those that are actually employed in common image texture processing are only a restricted number. Furthermore, note that the direction d can be computed in two way $+d$ or $-d$. If this distinction is not

taken into account the resulting matrix will be symmetric. In [137], that is likely the most famous work on this argument, where are proposed the so called *Haralick features*, it's also been suggested to use $d \in \{1, 2\}$ and $\theta \in \{0, 45, 90, 135\}$, respectively expressed in pixel units and angle degrees. Another commonly adopted way is simply to average all values obtained over different direction.
Dividing the $h_{d,\theta}(i, j)$ by the total number of contributions, carry out an estimate of the joint probability $p_{d,\theta}(i, j)$. The choice of the parameter d may affect the discrimination capabilities; smaller d has to be preferred in dealing with fine textures that instead achieves lower performance when they are coarser.
As for the characterization by histograms and maybe worst than that, it is unfeasible using directly the co-occurrence matrix, both for reasons of generalization capabilities than for its actual dimension. In fact it can be noted that without compressed representations, the co-occurrence matrix dimension grows with the square of the image value cardinality.
However from the co-occurrence normalized matrix some basic measures, or features, can be extracted. They may be essentially grouped in three sets:

1. *Statistic features* - are the usual basic statistical measures;
2. *Contrast features* - related to internal similarity properties;
3. *Orderliness features* - extracted from different image moments.

Let $p(i, j)$ the co-occurrence matrix, with $i, j \in [0, \ldots, G-1]$ the principal used measures are: *Mean (1st group):*

$$\begin{cases} \mu_i = \sum_{i,j=0}^{G-1} ip(i, j) \\ \mu_j = \sum_{i,j=0}^{G-1} jp(i, j) \end{cases} \tag{4.13}$$

Variance (1st group):

$$\begin{cases} \sigma_i^2 = \sum_{i,j=0}^{G-1} p(i, j)(i - \mu_i)^2 \\ \sigma_j^2 = \sum_{i,j=0}^{G-1} p(i, j)(j - \mu_j)^2 \end{cases} \tag{4.14}$$

Correlation (1st group):

$$\sigma_{i,j}^2 = \sum_{i,j=0}^{G-1} \frac{(i - \mu_i)(j - \mu_j)}{\sigma_i \sigma_j} \tag{4.15}$$

(with $\sigma_i = \sqrt{\sigma_i^2}$ and $\sigma_j = \sqrt{\sigma_j^2}$ the corresponding standard deviations)

Absolute value or Dissimilarity (2nd group):

$$AV = \sum_{i,j=0}^{G-1} |i-j|^2 p(i,j) \tag{4.16}$$

Inertia or contrast (2nd group):

$$I = \sum_{i,j=0}^{G-1} (i-j)^2 p(i,j) \tag{4.17}$$

Inverse difference or homogeneity (2nd group):

$$ID = \sum_{i,j=0}^{G-1} \frac{p(i,j)}{1+(i-j)^2} \tag{4.18}$$

Angular II moment (3rd group):

$$ASM = \sum_{i,j=0}^{G-1} p(i,j)^2 \tag{4.19}$$

Entropy (3rd group):

$$H = -\sum_{i,j=0}^{G-1} p(i,j) \log_2 p(i,j) \tag{4.20}$$

Max probability (3rd group):

$$MAX = \max_{i,j} p(i,j) \tag{4.21}$$

Note that some of this statistics may strongly degrade as the number of image intensity levels increase. A quantization step before is desirable in many situations. In [138] there is a good deep analysis about.
The previously presented quantities are the baseline and others might be extracted or derived for them.

4.2.3 Local Binary Pattern

Local Binary Pattern (LBP) are a method for texture description that is able to both combine statistical and structural approaches. First studies have been addressed in [139] where authors introduced a texture analysis based on simple *texture units*. The idea is

that each unit can be represented by a small number of elements ($\boldsymbol{e}$) each one taken from a small set of values ($[e_{min}, e_{max}]$). A vector is computed for all texture pixel considering a square neighbourhood (N). In this way each pixel has associated a value that represent it. The texture is then described by considering the distribution of these values over a patch. In this sense it is a combination of *structural* and *statistical* approaches.
Different implementations may vary in how all these values are computed, as for example one of the earlier study for each element a neighbourhood of 3×3 (i.e. $|\boldsymbol{e}| = 8$) was used with values taken in $[0, 2]$, and giving rise to a 3^8 different texture units. LBP is a non-parametric descriptor, close to the *census transform* [140], non-parametric local transforms relying on relative ordering of pixel intensities and not on the specific values. This leads to achieve some invariance to monotonic transformations of the intensity function.
Overall, one of the most successful method named *Uniform LBP* is reported in [9] and is the main reference for the present work. It is related to previous basic approaches on local binary pattern as [141] [142] and [143]. Once proposed, this method received much attention, that continues also in recent times (e.g. [144], [10], [145] and [146]). Before discussing the *Uniform* version of LBP it must be discussed the basic idea behind it. LBP was born to handle a single channel gray-level image even if can be easily extended to deal with different colour spaces. Starting from an entire image or a single patch for each pixel a neighborhood around it is selected to calculate its LBP value as in Figure 4.2. This value is computed considering a $N \times N$ window around the

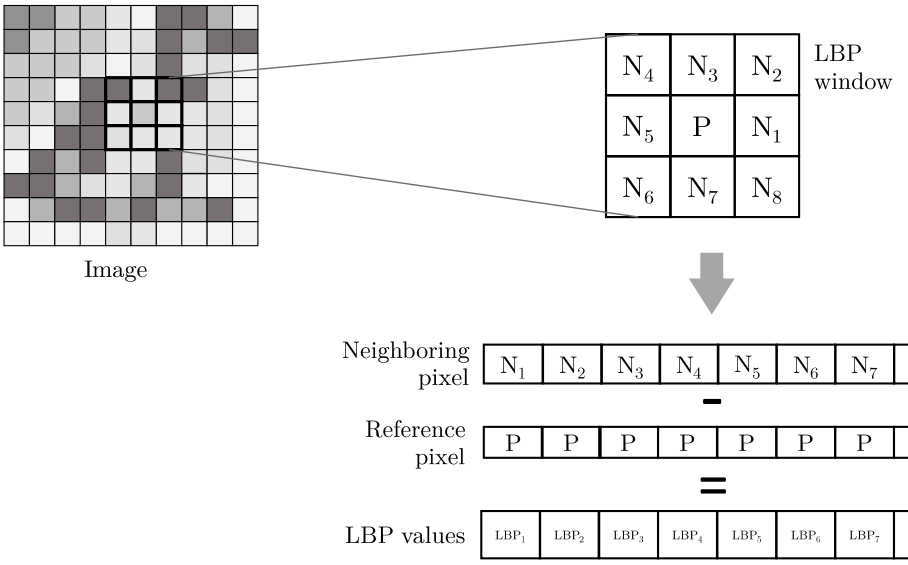


FIGURE 4.2: LBPs are computed for all image pixel considering their neighborhood. The value of the central pixel is then compared one-by-one with all adjacent ones.

central reference pixel p and each neighbouring pixel N_i is compared one-by-one with the value of p. The complete version of *Uniform LBP* has the properties to be invariant about monotonic transformation of pixel values and against image rotations. It starts to consider a monochrome textured square patch T, composed by a number of P (with $P > 1$) pixels. Let

$$T = t(c, g_0, \ldots, g_{P-1}) \tag{4.22}$$

where c is the value of central pixel and g_i, with $i = 0 \ldots P-1$, are the values of neighbouring pixels. Differently from previous methods this neighbourhood is composed by equally spaced pixels at a radial distance R ($R > 0$) expressed in pixel units, as it is represented in Figure 4.3 with different R and P values. Each pixel in the position p

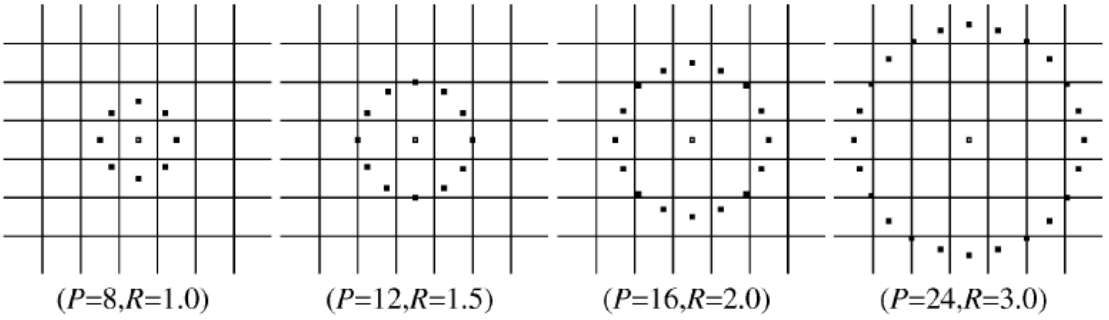


FIGURE 4.3: The circular neighbourhood of LBP for different value of radius R and pixels P. (Image from [9])

has coordinates

$$(-R \sin \frac{2\pi p}{P}, R \cos \frac{2\pi p}{P}) \tag{4.23}$$

and for a practical use these values might be obtained by interpolation. The local binary pattern of each central pixel of the texture patch is calculated by subtracting its value c to all neighbouring pixels of value g_i. The texture T is hence defined as

$$T = t(g_0 - c, g_1 - c, \ldots, g_{P-1} - c) \tag{4.24}$$

and this representation achieves an invariance against diffuse changes in luminance that rigidly interest all the texture. Theoretically the invariance is acquired by value shifts that might occur uniformly over all pixels (Figure 4.4).

This invariance has obviously a price in terms of lost information. Anyhow this is not a big deal because we theoretically consider textures that have equal luminance change and, excluding some kind of noise and considering patches with small R values, it is reasonable that the luminance uniformly vary.

It is possible to achieve invariance related to other gray-level transformations, as scaling or more in general for those due to a monotonic function. To realize this, all references

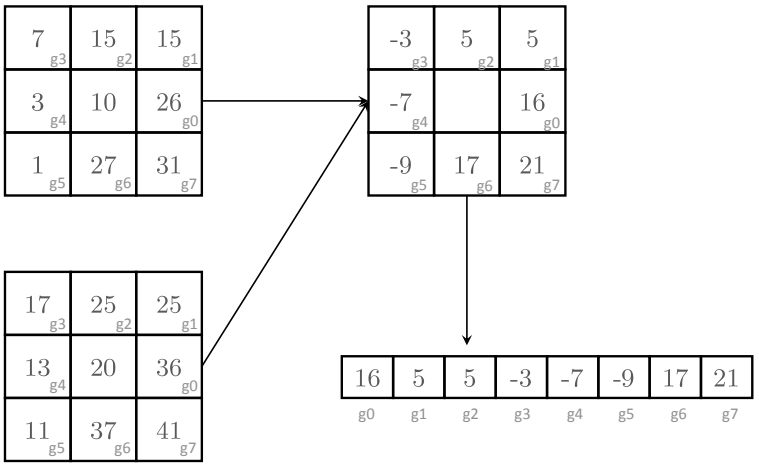


FIGURE 4.4: The effect of gray-level shift invariance.

to the actual pixel values are discarded. The central pixel of value c is compared with its neighbourhoods by the function,

$$s(x) = \begin{cases} 1 & if \quad x \geq 0 \\ 0 & if \quad x < 0 \end{cases} \tag{4.25}$$

that is nothing more that the $sign(\cdot)$ operator and T can be represented as:

$$T = t(s(g_0 - c), s(g_1 - c), \ldots, s(g_{P-1} - c)). \tag{4.26}$$

Now each pixel comparison give rise to a single bit $(0-1)$ of information depending if the neighboring pixel is respectively less or greater than the reference one. Having P pixel in the circular neighborhood as output of this transformation we obtain an ordered binary vector of length P (see Figure 4.5). It is this step that substantially determines the name of *binary pattern*. The ordered processing of neighbouring pixel is crucial. Usually the binary vector is constructed starting from the first (most significant) position and pixels are taken starting from the right one in anticlockwise. We can express the *local binary pattern* of a pixel considering a circular neighbourhood with a radius R and composed by P elements as:

$$LBP_{P,R} = \sum_{i=0}^{P-1} s(g_i - c)2^P. \tag{4.27}$$

With a neighbourhood pixel set composed of P elements the binary pattern has 2^P possible configurations. For now this local binary pattern operator differs from earlier version (like in [141]) only for using a circular neighbourhood. The LBP can be further

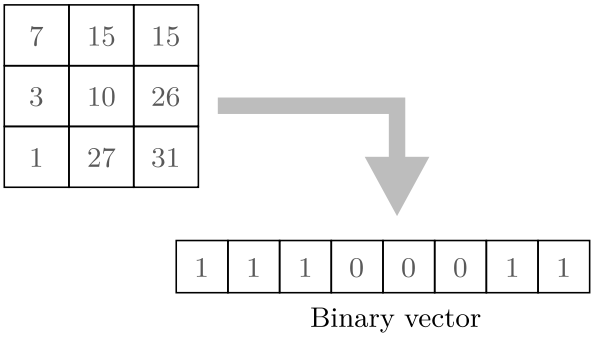


FIGURE 4.5: The comparison with neighbouring pixel give rise to an ordered binary vector.

extended to handle invariance against rotation.

In fact, comparing two $LBP_{P,R}$ extracted from corresponding point of two images that are identical unless a rotation factor, these can lead to a very different binary pattern and a recognition process are likely to fail over them. Depending on the circular neighbourhood sampling angle there are exactly P different rotation for an $LBP_{P,R}$ equally spaced by $\frac{2\pi}{P}$ degree (clearly the hypothesis is that the circular neighbourhood sampling is uniform). The easiest way to handle with differences due to rotations is to take all the possible shifting of an $LBP_{P,R}$. As drawback, this solution will increase the computational cost about P times during comparisons between $LBP_{P,R}$.

A second way to achieve rotation invariance is to define a $LBP_{P,R}$ normalization over all possible P rotations. Hence the new definition for the binary pattern is:

$$LBP^{ri}_{P,R} = \min\{ROR(LBP_{P,R}, i) \quad | \quad i = 0, 1, \ldots, P-1\} \tag{4.28}$$

where $ROR(\cdot)$ is a right shift binary operator. This leads the LBP operator to loose more allowed configurations and so a certain amount of discrimination power.

Figure 4.6 shows the 36 possible unique binary pattern, calculated with $P = 8$, that have the property of rotation invariance. Except for the patterns reported in the first rows—those from #0 to #8 in Figure 4.6—all the others, from the second to the fourth row, have not a straight relation with common texture structures. Patterns from #0 to #8 can have, instead an interpretable meaning. The *all-0* and *all-1* configurations (respectively #0 and #8) may stand for uniform or spotted texture element. The others patterns (#1 to #7) may instead represent simple component of edges or straight line. Experiments conducted in [142] with $LBP^{ri}_{P,R}$ showed that using the entire set of 36 rotation invariant LBP may be often useless or even counterproductive. In particular the authors of this work found that over the 90% of $LBP^{ri}_{P,R}$ patterns belong to the types from #0 to #8. This reduced set of 9 elements is so the only one used in most

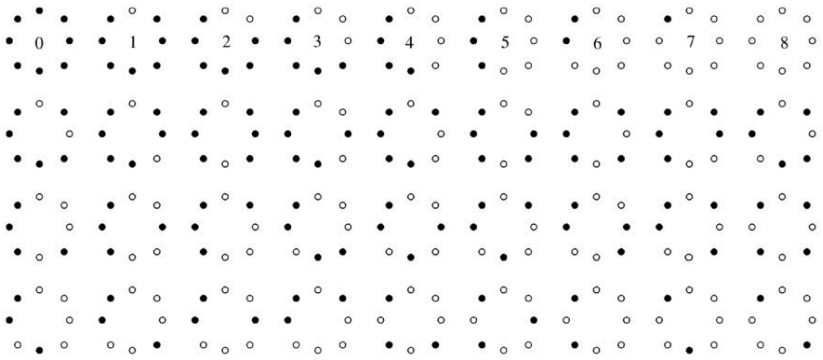


FIGURE 4.6: The possible rotation invariant binary pattern configurations for $LBP^{ri}_{P,R}$. (Image from [9])

practical applications and its elements are also known as *uniform LBPs*.
Mathematically to define a *Uniform Local Binary pattern* a measure $U(\cdot)$ is needed to be defined. This operator measures the number of transitions that occurs in the P-length vector of each $LBP_{P,R}$. The 9 uniform patterns (Figure 4.7) have a $U(\cdot)$ value that is

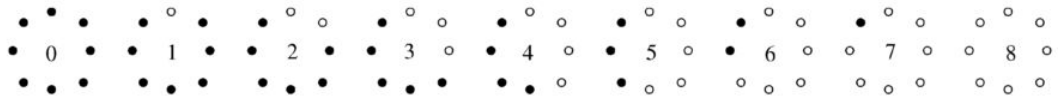


FIGURE 4.7: The *uniform* patterns. (image from [9])

less or equal to 2.
In conclusion the *Uniform Local Binary Pattern* can be expressed (following the notation reported in [9] as:

$$LBP^{riu2}_{P,R} = \begin{cases} \sum_{i=0}^{P-1} s(g_i - c) & if \quad U(LBP_{P,R} \leq 2) \\ P+1 & if \quad U(LBP_{P,R} > 2) \end{cases} \tag{4.29}$$

with $U(\cdot)$ expressed as:

$$U(LBP_{P,R}) = |s(g_{P-1} - c) - s(g_0 - c)| + \sum_{i=1}^{P-1} |s(g_i - c) - s(g_{P-1} - c)|. \tag{4.30}$$

All the "non-uniform" patterns are grouped in a single class, so, considering $P = 8$ and $R = 1$ all possible configurations of LBP binary vector belong to one of 10 final configurations.
The choice of neighbourhood size P and radius R are obviously related. As R increase the circular quantization angle might be reduced and P increased. For a small number of neighbourhoods the interpolation function, used to realize the circular sampling, can be avoided. Dealing with usual natural images instead of using bigger values of R can

be convenient—without considering particular application fields—to reduce the image scale and then use values as $(P, R) \in \{(8, 1), (16, 2)\}$.
To summarize, $LBP^{riu2}_{P,R}$ is the "full" version of the *Uniform LBP* and is theoretically able to deal with monotonic value transformations and image rotations.
Remembering that each of these $LBP^{riu2}_{P,R}$ invariant properties imply an information leak, after some preliminary experiments we adopted a slightly different approach consisting in the uniform local binary pattern without the rotation invariance. We refer these as $LBP^{u2}_{P,R}$ and a similar approach is also discussed in [10].
The $LBP^{riu2}_{8,1}$ maps all patterns in 10 possible configuration—9 uniform plus 1 non-uniform. Removing the rotation invariance property and keeping the *uniform pattern* concept, there is the necessity to (re)consider all the possible configurations due to image rotations. As previously discussed each rotation of a $2\pi/P$ angle corresponds to a vector shifted of P positions. In Figure 4.8 are reported all the possible combinations of binary vector that generate the pattern set. Considering that each vector—with equal or less

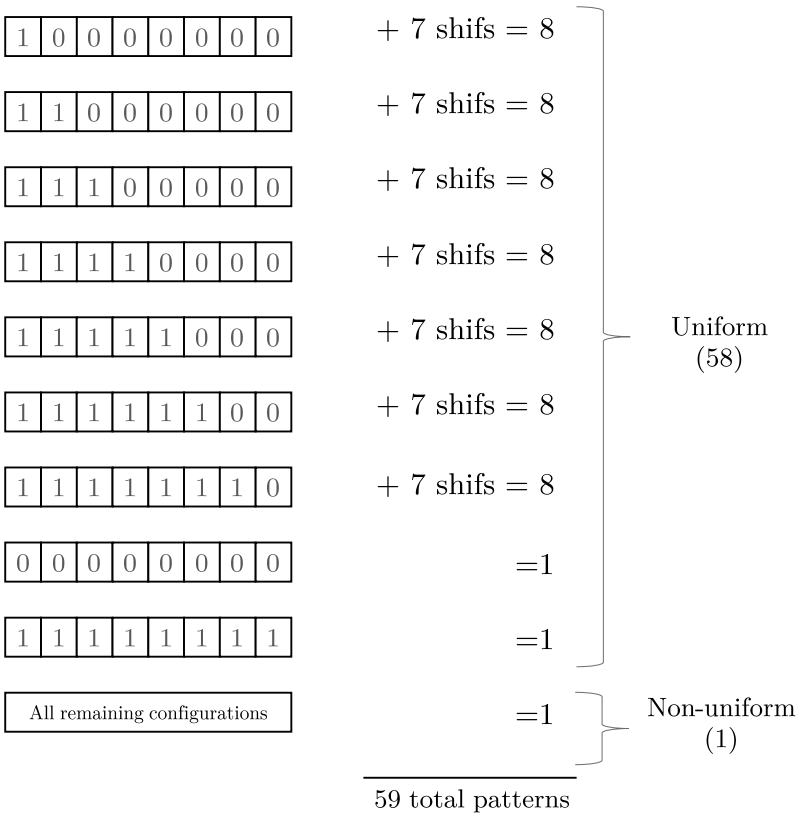


FIGURE 4.8: All the possible binary vectors achieved with a parameter $P = 8$).

than two 01/10 transitions—has 8 possible shifts and that the *all-0* and *all-1* vectors are shift invariant, in total there are $2 + 8 \times 7 = 58$ uniform patterns.

All the possible circular pattern configurations are visually expressed in Figure 4.9.

To these 58 patterns must be added one, non-uniform, configuration that represent all

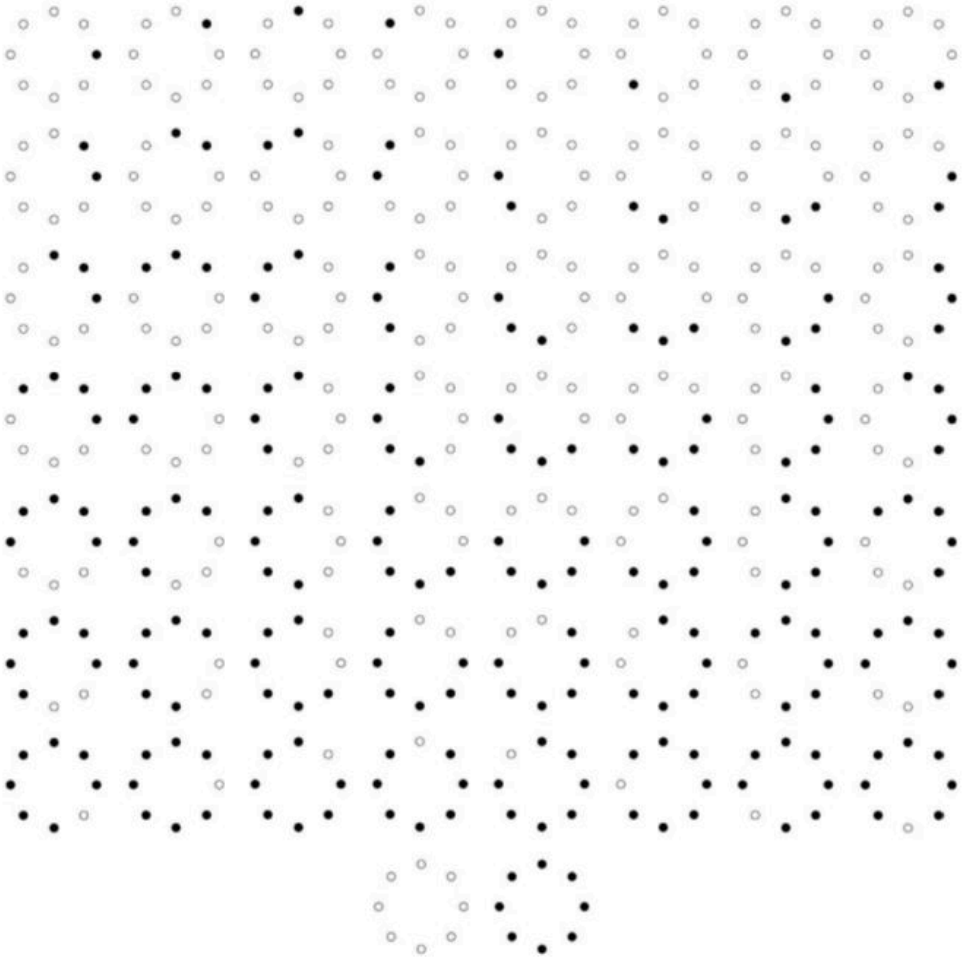


FIGURE 4.9: Visualization of the 58 uniform binary patterns $LBP^{u2}_{8,1}$.(from [10])

the configurations that have three or more 0-1 transitions in the binary vector. This means that all the non-uniform configurations are treated equally. So, in total the $LBP^{u2}_{8,1}$ has 59 different configurations.

A non-trivial textured image patch is made up of at least dozens of pixels. Local Binary Pattern processes, in the previously described manner, each pixel at time (Figure 4.10) and associate it to one of the previously defined patterns. Like many windowed filters, to deal with the border of a patch that have not a complete neighbourhood many approach can be employed, as predefined values circular border extensions or border repetitions. The most simple (and used) approach is to not consider all these pixels without all neighbouring elements.

Histograms are employed to finally describe a texture patch with uniform LBPs. A bin corresponds to each LBP and the feature vector describing an image patch has a size equal to the number of pattern. Then, in the case of $LBP^{u2}_{8,1}$ the feature vector size

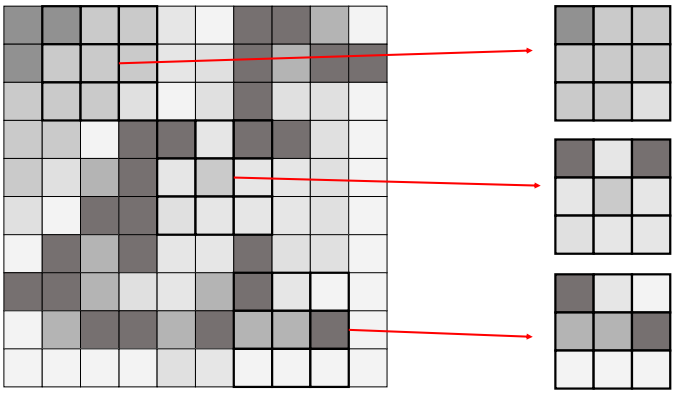


FIGURE 4.10: LBP extraction from texture patches.

is actually 59. It must be underlined that without the uniform pattern selection this dimension would be 2^8 and, in general, increasing the neighbourhood size P, this will increase the feature vector size.

4.3 Datsets

To conduct our experiments we noticed that the availability of commonly-adopted underwater datasets is very poor. The majority of works in this field are mostly based on using simulations or task-specific dataset. The use of pertinent dataset is anyhow, a key issue in every classification task (see [147]).

Thanks to our participation in the project ARROWS (see section 2.2) we had the chance to collect several hours of underwater videos taken both from vehicles and divers. We arranged data in approximatively ten original dataset, each one built regarding some common environmental and acquisition characteristics that we meant to investigate. Alongside them we placed other two dataset made by publicly available images after an opportune preprocessing.

Every dataset—derived generally from one or two video sources—is divided in a number of classes each one consisting in a collection of several image patches.

Not all environments can show or have representatives for each class and for this reason the class-composition is properly one distinction point between our datasets. Furthermore it must be underlined that also if they share the same name, classes might appear extremely different according the dataset to they belong.

All the labelling procedure was carried out by hand. In particular, we created a semi-automatic software to preliminary segment each images according to some basic measures. From these patches we selected those clearly representing areas (i.e. classes) of

interest and we labelled them. This allowed us to collect thousands examples for every class but for a number of hard datasets we had to choose and pick up patches totally by hand.

Table 4.1 summarize the twelve dataset that we employ and their class splitting.

The class names that we used are self-describing and intentionally represent a broad

TABLE 4.1: Main datasets employed for classification and their composition.

Id	**No. of classes**	**Class 1**	**Class 2**	**Class 3**	**Class 4**	**Class 5**
D01	5	algae	coral	h.vegetation	sand	vegetation
D02	5	algae	archaeo	sand	vegetation	water
D03	3	rock	sand	vegetation	-	-
D04	2	sand	vegetation	-	-	-
D05	3	backgrnd	group	sand	-	
D06	4	algae	archaeo	sand	vegetation	-
D07	4	algae	archaeo	sand	vegetation	-
D08	4	algae	archaeo	sand	vegetation	-
D09	3	algae	archaeo	vegetation	-	-
D10	2	coral	sand	-	-	
D11	3	rocks	sand	vegetation	-	-
D12	3	archaeo	sand	water	-	-

human-like description. In this work the focus is firstly on labelling areas, not objects or particular shapes. For example the *archaeological* class stand for a generic object category and not a specific kind. Each dataset may be characterized by a different type of archaeological objects.

Our classification task is not meant to work at an object level but is looking for areas where they may be present in group. At this regard, we needs to underline that looking for isolated and well-specific archaeological things might be, in the general case for underwater scenario, an extremely hard task. In fact, after some time, objects in an underwater environment tend to fuse with the environmental appearance, their shapes are modelled by the encompassing atmosphere and so their retrieval difficulty become higher with time.

In the following, a brief description and some examples for everyone of this twelve dataset is provided. With the *ARROWS Project* label we refers to original material or activities related somehow with this project.

- Dataset D1

  Location: Tasmania.

  Type: Natural environment.

  Recorded by: AUV.

  Classes: algae, coral, high vegetation, low vegetation, sand.

  Source: Acknowledgement to the Australian Centre for Field Robotics.

This dataset was made starting from the *Tasmania Coral Point Count* data, used also in [148] and [149]. From these we takes only a subset of selected images and they have been classified according to classes that we had previously isolated. As it is shown on 4.11, images are characterized by a good definition, clarity and with low colour distortions.

- Dataset D2
  Location: Elba island.
  Type: Natural environment.
  Recorded by: Human.
  Classes: algae, archaeological finds, vegetation, only water, sand.
  Source: ARROWS and THESAURUS projects.
  This dataset was recorded by humans at low deep and characterized by the presence of human made objects that simulate archaeological finds and the presence of a big wreck. As shown in Figure 4.12, images are clear, environmental colours tend to green and there is an evident sunlight illumination scattered by the sea surface. The seabed appearance is largely uniform.

- Dataset D3
  Location: Israel.
  Type: Natural environment.
  Recorded by: AUV.
  Classes: rock, sand, vegetation.
  Source: ARROWS project.
  This dataset (Figure 4.13) was recorded by AUV with a tilt angle of approximatively 45 degrees. The seabed is characterized by the presence of a mixture of vegetation and sand. The image clarity is not uniform, colours are affected by high distortion and sufficient sunlight illumination. All these aspect vary in accordance to small depth changes during vehicle navigation.

- Dataset D4
  Location: Israel.
  Type: Natural environment.
  Recorded by: AUV.
  Classes: low vegetation, sand.
  Source: ARROWS project.
  Even if recorded in a different place and different depth, this dataset is close to the D3 dataset. In this dataset (Figure 4.14) there are not areas of particular interest—in fact we used it as a 2-class dataset– but colour distortions and blur effects assume here notably values. The sunlight presence and the camera settings

used have contributed to record images characterized by both dark and saturated parts.

- Dataset D5
  Location: Indoor pool.
  Type: Artificial environment.
  Recorded by: Human.
  Classes: pool background, grouped objects, some sand
  Source: ARROWS project.
  This dataset has been created to initially test our algorithms in a controlled environment. Classes are in this case slightly different from other datasets (Figure 4.15). Image resolution was limited and colours are approximatively natural.

- Dataset D6
  Location: Sicily.
  Type: Natural environment.
  Recorded by: AUV.
  Classes: algae, vegetation, archaeological finds, sand.
  Source: ARROWS project.
  Images recorded by AUV during a Sicily campaign. Other than classes tipycal of a natural underwater environment (vegetation and sand) there is a wide presence of groups of archaeological vessels that as may be seen on Figure 4.16 have an appearance not ever easily discernible from the seabed. This fact is also due to a strong image colour distortions. The appearance is almost clear and a little bit of blur effect is sometimes due to the vehicle motion over the seabed.

- Dataset D7
  Location: Sicily.
  Type: Natural environment.
  Recorded by: Human.
  Classes: algae, vegetation, archaeological finds, sand.
  Source: ARROWS project.
  This dataset 4.17 is similar, in content, to the D6 and mostly D8. They share—other than the originating area—the same classes. What is different is the way in which images have been registered. From the dataset D6, it changes the dominant colour that now is green. Images are clear e with a low blur. Light entirely comes from the sunlight illuminating the scene in a uniform manner.

- Dataset D8 *Location:* Sicily.
  Type: Natural environment.

Recorded by: Human.
Classes: algae, vegetation, archaeological finds, sand.
Source: ARROWS project.
Dataset close to the D7 one. Compared to this latter changes affect a little bit the environment, the depth (now objects are closer to the camera), the illumination (slightly higher) and the camera-induced optical distortion. Image pureness is comparable and the dominant colour is still a brighter green (see Figure 4.18).

- Dataset D9 *Location:* Sicily.
  Type: Natural environment.
  Recorded by: Human.
  Classes: algae, vegetation, archaeological.
  Source: ARROWS project.
  This is the last (of four) Sicilian dataset (Figure 4.19). The environment looks again similar but now camera, illumination and mostly colours are significantly different. Due to the motion there is more blur on image and higher is the haze level (also due for a certain amount to the camera settings). Colours are now more realistic with less distortions than other cases. Images are taken very close to the seabed and there is a strong presence of vegetation.

- Dataset D10
  Location: Pacific ocean and Israel.
  Type: Natural environment.
  Recorded by: AUV.
  Classes: coral, sand.
  Source: ARROWS project.
  This is a 2-class dataset (see Figure 4.20) created from images of two previous dataset, D1 and D4. This is an experimental dataset and its purpose is to test classifiers about two extremely different environment. One (D1) clear and rich in colour and vegetation, the other (D4) characterized by distorted colours, haze presence and a larger scale. Clearly the two classes are mixed together starting from the corresponding ones originating from the two datasets.

- Dataset D11
  Location: Elba Island.
  Type: Natural environment.
  Recorded by: AUV.
  Classes: rock, sand, vegetation
  Source: THESAURUS project and *Soprintendenza Archeologia della Regione Toscana (N.O.S.)*

This dataset (Figure 4.21) originates from an environment with high presence of archaeological finds. The camera on this dataset is kept with optical axes perpendicular to the seabed so images are low distorted and the relative slow velocity of the vehicle allow clear acquisitions. Colours tents to green, but illumination is not uniform due to an artificial spotted light.

- Dataset D12
  Location: Elba Island.
  Type: Natural environment.
  Recorded by: AUV.
  Classes: archaeological finds, sand, water.
  Source: THESAURUS project and *Soprintendenza Archeologia della Regione Toscana (N.O.S.)*
  This dataset (Figure 4.22) is taken in an area close to the one of dataset D11, but with totally different settings. The vehicle has the camera on the front and it is pointed almost horizontally and parallel to the seabed. This causes an image with a huge depth. The haze, interacting with floating particles, increases with the distance so details and colours are distinguishable only for close areas in where the lights act. Nevertheless this dataset is interesting because the high presence of many archaeological finds.

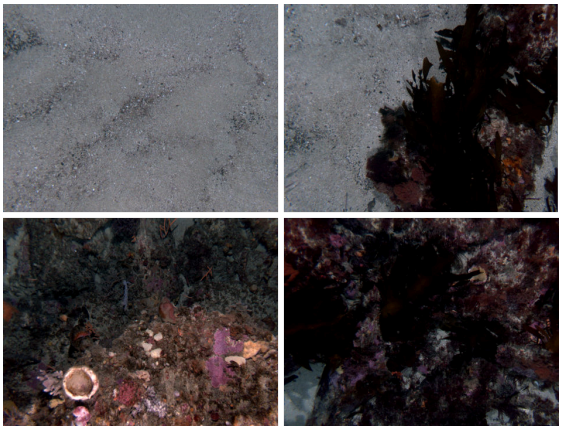


FIGURE 4.11: Dataset D1

FIGURE 4.12: Dataset D2

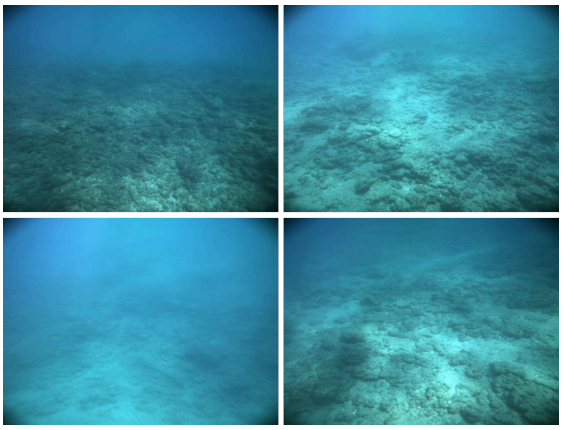


FIGURE 4.13: Dataset D3

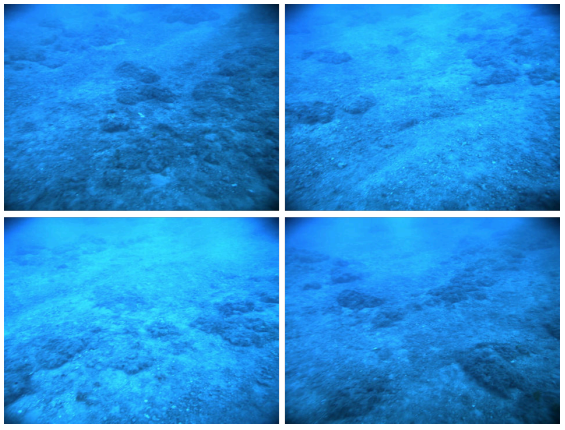


FIGURE 4.14: Dataset D4

FIGURE 4.15: Dataset D5

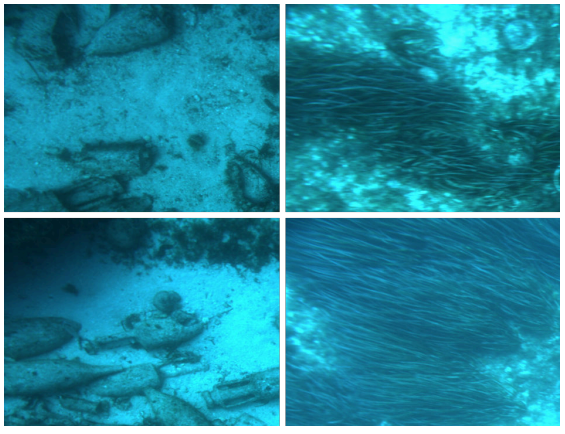


FIGURE 4.16: Dataset D6

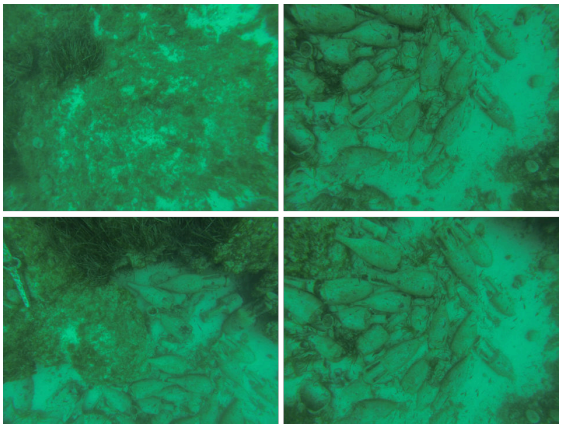


FIGURE 4.17: Dataset D7

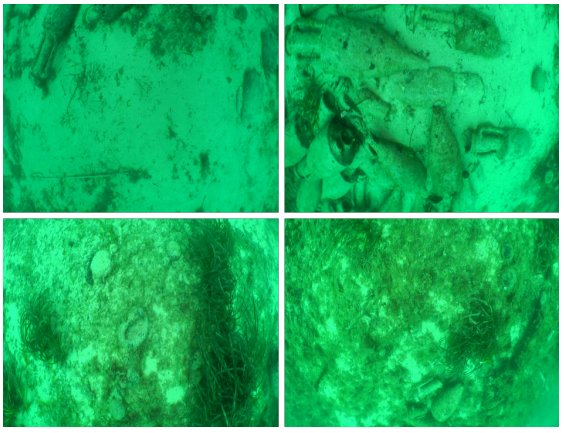


FIGURE 4.18: Dataset D8

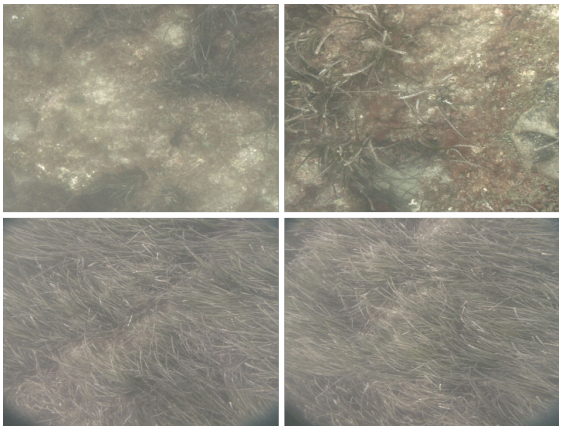


FIGURE 4.19: Dataset D9

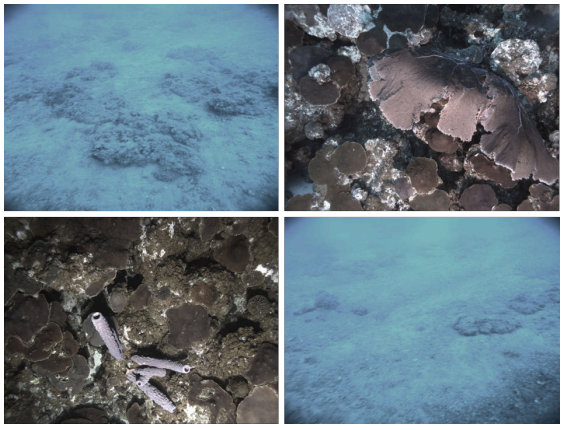


FIGURE 4.20: Dataset D10

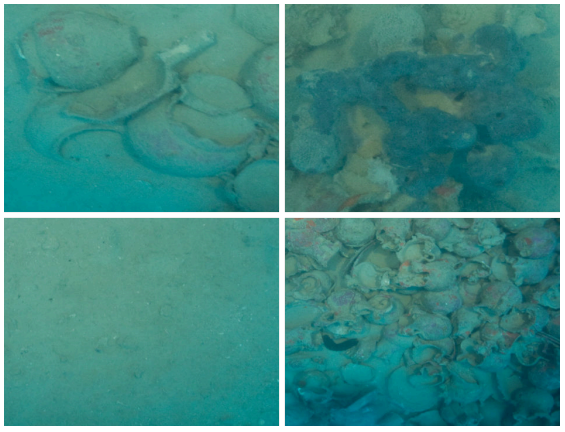


FIGURE 4.21: Dataset D11

FIGURE 4.22: Dataset D12

4.4 Experiments

We evaluate the performances of the three previous defined sets of features to classify the underwater scenario. These feature sets was selected because from a theoretical point of view about underwater environmental characteristic and after some preliminary evaluation test they might express the best performance.
This section describes in detail both the feature sets and the classification scheme actually adopted to conduct our tests.

4.4.1 Features

As discussed in chapter 1.1 the underwater scenario including all living and non-living things— and without considering particular areas—is today still an almost totally pure and natural environment. Places characterized by a massive human presence are in general likely denoted by more regular shapes (like streets and buildings) and discontinuity in their appearance that can aid in recognition and classification tasks. The natural scenario vary in a continuous manner presenting similar but not identical structures. Statistics can in general better handle with the variation of natural images. Each environment has its own characteristics and thus its own statistics ([150] and [151]). Statistics can be extracted from image histograms ([152], [153] and [154]), that are able to synthesize images totally discarding the spatial order within it.
Using the first order statistics to characterize underwater scenarios is one of the simplest and straightforward approaches. These features can describe the general appearance of the scenario, with a certain amount of invariance from scale variations and environmental conditions.
To this end two things have to be kept in mind. Underwater scenario usually does not permit to take picture with a wide field of view—because light might be strongly attenuated—so the scale of images taken from AUV or ROV haven't generally strong scale-space variations. Secondly we can not trust on colours. In fact, depending on water conditions there might be high amount of alterations.
To construct histogram from an image patch, colour channels can be used individually or combined together. The straight solution is to look at the gray-level values because depending on the particular scenario green or blue channel can predominate (e.g. Figure 4.23).
Some algorithm to normalize the colour appearance might be sometimes adopted.
Drawbacks of using first order statistics are mainly due to the fact that they do not hold information about neighbouring pixels.

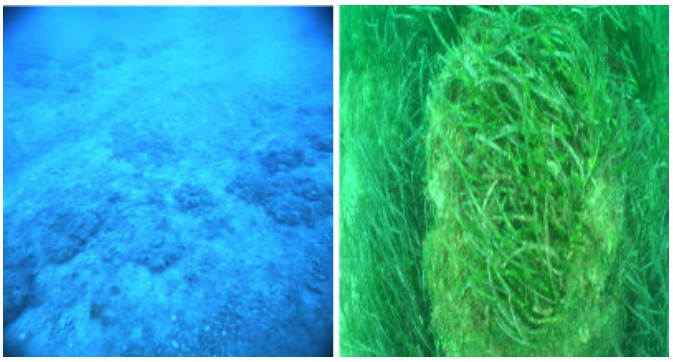


FIGURE 4.23: Example of underwater images both dominated from a single colour, blue (left) and green (right).

An extension in this direction is the use of second order statistics method for texture analysis. This approach starts from a *Gray-Level Co-occurrence Matrix* (GLCM) or one of its derivation. Handling with spatial relationships may lead to have more discrimination power but obviously it determines in general a price that has to be payed in terms of scale space invariance and noise sensitivity. The *uniform Local Binary Pattern* is a descriptor that both combine a structural and statistical approach (see section 4.2.3) and can suit the problem of underwater environment classification. Image patches are still described using histograms, but instead of representing straightly statistic of tonal value, they represent the distribution of a set composed by 59 (in case of $LBP^{u2}_{8,1,}$) known pattern. Practically we aim to compare how tacking into consideration some primitive local pixel configurations, their distribution can describe and discriminate a texture. In comparison of, other methods LBP are more sensitive in resolution changes, so in practical applications, where the fields of view can greatly vary, approaches tacking with Gaussian pyramid representation may be employed (see [155]).
Experiments here presented compare performance over a wide range of underwater scenarios evaluating primarily which features—from those proposed—achieve better results if used in a classification tasks. The chosen approach is a *dense sampling* that appeared more reasonable in our scenario where the entire underwater scene is relevant.
These features are largely employed with success in many classification task as satellite image processing or biological microscopy, both fields where texture analysis plays a predominant role. We think that underwater scenario has a lot in common with these type of analysis. What makes interesting our investigation is to overcome issues induced by the underwater environment which is, to our knowledge, not researched as well.

4.4.2 Classification method

The three texture description methods previously presented (section 4.2), are based on approaches that allow a deep and significant comparison. Each one of these features may share the same architecture for classification. Due to the lack of widely adopted underwater-specific dataset we created our ones (section 4.3) with the intent to collect a large range of examples and environmental scenarios.
The first part of our work was to preliminary test performance of various potential classification and segmentation methods, both supervised and unsupervised. After tests regarding the *K-Means*, *Nearest Neighbour* ([156]) and other *semi-supervised* machine learning approaches ([157]), we drove our attention to *Support Vector Machines* ([158]) and we used it for our experiments.
Support Vector Machines (SVMs) are a widely employed machine learning method ([159]). Even if this work is focused on classification, it can used also in problems of regression and novelty detection. The initial formulation takes into consideration the 2-class problem, looking for the *optimal* separating hyperplane (in the original or a transformed feature space). In SVM the decision boundary is chosen to be the one for which the margin is maximized, i.e. those that minimize the distance between the decision boundary and any of the samples in the training set. More details about the theory behind SVM can be found on [160].
Other than by preliminary evaluations, the choice of SVM has been suggested also by the following considerations:

- a supervised approach better fit our problem of classification with a priori known classes
- SVMs are a mature and well studied method and they are already used in many classification problems
- the SVM behaviour can be controlled by a lot of well-known parameters and its outputs may be easily understood
- once trained, SVM classifiers are sufficiently fast to process new examples and so can be implemented to deal with real-time tasks
- SVM, in addition to other approaches, may provide acceptable results in case of relative small sets of training examples. On the other hand, Bayesian approaches (e.g. *Naive Bayes* [161], [162]) and more in general *generative* probabilistic approaches may need a greater number of examples to work well for this kind problems

- there are available, complete and fast libraries providing good and efficient implementation of SVMs.

Classification task with SVM is achieved following the architecture shown in Figure 4.24.
Being a supervised approach it is composed of two distinct phases; *learning* and *clas-*

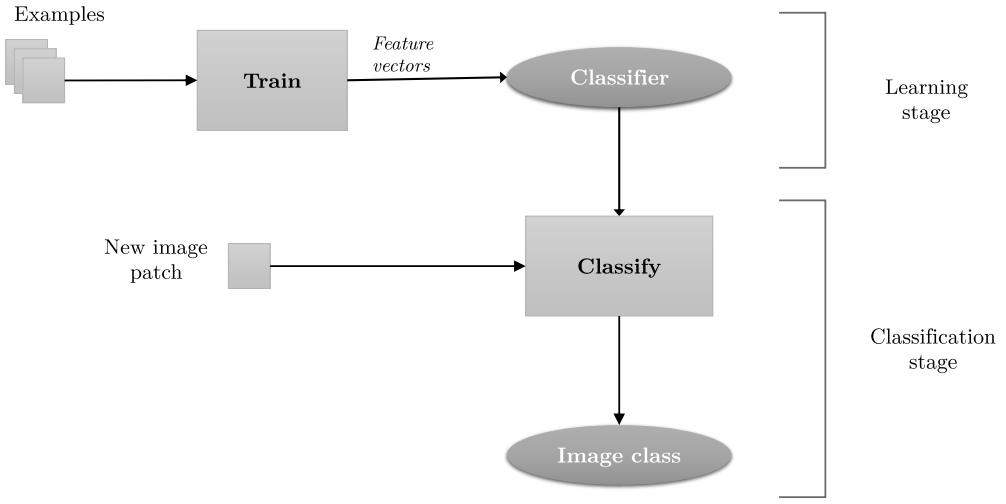


FIGURE 4.24: The adopted schema for SVM classification.

sification. In the learning phase we feed the SVM with examples (i.e. feature vectors) previously extracted for every image patch. During learning, the algorithm was feed with examples and the associated class label to they belong.
In this work we trained SVM according to various configurations and parameters. The most relevant are related to the kernel type and regularization. Secondary settings are further related to other details as for example the adopted stopping criterion, but for now they are less relevant.
Steps for doing classification are practically the same than training. Feature vectors are extracted from the image and processed with the classifier previously learned. Although the two phases are separated, it is obviously mandatory that learning and classification parameters have to be consistent.
An SVM classifier has to be (re-)trained every time new examples need to be added. For this reason the dataset provided in the training step has to be a valid representation of what we want to classify.
As previously said, SVM was born to perform binary classification, dealing with only one class of *positive*(+) and one of *negative*(−) examples. Here our objective is a more general *multi-class* classification so we need to consider further extended version of the

original Support Vector Machines ([163], [164]). Multi-class classification problems can be carried out by SVM in basically two way (see [165]): the *One-vs-All* and the *One-vs-One* approach.

The first is the earliest adopted and the most theoretically straightforward schema [166]. For K classes (with $K > 2$) it trains K models, one for each class. Examples in the i class have a positive label while all the other a negative one. A number of K SVM models means to have K decision functions ($d_i(\cdot)$). Considering an example $\boldsymbol{x}$ it will classified according the C class such that:

$$C(x) = \underset{i=1...K}{argmax} \quad d_i(\boldsymbol{x}) \quad . \tag{4.31}$$

In other words $\boldsymbol{x}$ is assigned to the class corresponding to the decision function with the largest value.

The second multi-class approach to SVM—and *de facto* mainly used in this work— is the *One-vs-One* [167][168]. Now, a number of $K(K-1)/2$ classifiers is trained, each one with data (i.e. examples) of only two single classes. The class that emerges in the highest number of one-by-one binary classifications will be chosen as the class of the given example $\boldsymbol{x}$. Formally, let $d_{i,j}$ the decision function discriminating the class i from the class j (respectively the positive and the negative one). Considering the input example $\boldsymbol{x}$, if $d_{i,j}$ votes for positive the class i get a vote; otherwise (negative response) is the j class which is incremented of one vote. When all binary decision functions have been applied, the example $\boldsymbol{x}$ is assigned to the class C with the largest amount of votes. In [165] was particularly suggested this latter approach (one-vs-one) as a good choice for practical use.

4.4.3 Evaluation

To evaluate the classification performance the usually employed parameter is the *accuracy* (ACC) measure. It is calculated as:

$$ACC = \frac{No. \quad correct \quad predictions}{No. \quad total \quad predictions} \cdot 100 \quad . \tag{4.32}$$

This is the most straightforward measure to make comparisons between classifiers.

We argue that for general binary classification problems there are a lot of available measures—typical related to the information retrieval field—that can be used. In fact with only two class (positive and negative) there are a lot of significant measures that might be employed as: *F1-score*, *Sensitivity*, *Specificity*, *Precision*, *Negative Predictive*

Value, *Fall-Out*, *False Discovery Rate*, *False Negative Rate (missing)*, *Matthews Correlation Coefficient*. In [169] and [11] is reported a good discussion about these measures for binary problems.

As reported in Figure 4.25, some of the previously considered measures can be extended to the general case by averaging by class. Another possibility of performance analysis is

Measure	Formula	Evaluation focus
Average Accuracy	$\frac{\sum_{i=1}^{l}\frac{tp_i+tn_i}{tp_i+fp_i+fn_i+tn_i}}{l}$	The average per-class effectiveness of a classifier
Error Rate	$\frac{\sum_{i=1}^{l}\frac{fp_i+fn_i}{tp_i+fp_i+fn_i+tn_i}}{l}$	The average per-class classification error
$Precision_{\mu}$	$\frac{\sum_{i=1}^{l} tp_i}{\sum_{i=1}^{l}(tp_i+fp_i)}$	Agreement of the data class labels with those of a classifiers if calculated from sums of per-text decisions
$Recall_{\mu}$	$\frac{\sum_{i=1}^{l} tp_i}{\sum_{i=1}^{l}(tp_i+fn_i)}$	Effectiveness of a classifier to identify class labels if calculated from sums of per-text decisions
$Fscore_{\mu}$	$\frac{(\beta^2+1)Precision_{\mu}Recall_{\mu}}{\beta^2 Precision_{\mu}+Recall_{\mu}}$	Relations between data's positive labels and those given by a classifier based on sums of per-text decisions
$Precision_M$	$\frac{\sum_{i=1}^{l}\frac{tp_i}{tp_i+fp_i}}{l}$	An average per-class agreement of the data class labels with those of a classifiers
$Recall_M$	$\frac{\sum_{i=1}^{l}\frac{tp_i}{tp_i+fn_i}}{l}$	An average per-class effectiveness of a classifier to identify class labels
$Fscore_M$	$\frac{(\beta^2+1)Precision_M Recall_M}{\beta^2 Precision_M+Recall_M}$	Relations between data's positive labels and those given by a classifier based on a per-class average

FIGURE 4.25: Extension of some classic evaluation measures to the multi-class scenario. (table from [11])

the *ROC (Reception operating Characteristic)* [170], but as the previously measures for multi-class problem is not today a well developed field. In fact, for multi-class problem all these measures loose their main descriptive properties; for this reason and also for shortness they are not reported here, even if our classification framework actually computes some of that measures.

Nevertheless, what can be done is reporting the *Confusion Matrix* (or *contingency matrix*).

The confusion matrix can better summarize the performance of classification with multiple classes. It is a square matrix with $C \times C$ dimension where C is the number of class. By rows it reports the actual class and by columns the predicted ones. Let i and j respectively the row and column index; the cell in position (i, j) represents the amount of examples—expressed as absolute value or percentage—that belongs to class i and were predicted of class j. So, all correct classifications lie in the matrix main diagonal (i.e. $i = j$). From the confusion matrix may be extracted information about which are the most corrected classified classes or which are those that are often mistaken.

Using only the accuracy as a performance measure might sometimes be trivial. Accuracy assesses the overall effectiveness of a classifier but it suffer unbalanced datasets. This happens when there are more examples of one class instead another. For example in a two-class classification problem with 90 examples of class A and 10 examples of class B, one classifier that labels all examples as A-class will realize a 90% of accuracy, but this does not means that it is a good classifiers having 0% of recognition rate on B class. To overcome this issue it is a good choice to construct datasets with a good

balancing between the number of examples per class and this is what we have done in our experiments.

Chapter 5

Underwater classification: Results

Here are presented the obtained results in classifying image patches taken from underwater scenarios. The classification results was achieved evaluating the previously defined feature sets, in particular:

- First order statistics,
- Second order statistics,
- Uniform Local Binary Pattern.

All these descriptors were tested over all the datasets discussed in Section 4.3.
The SVM multi-class classification was conducted with the *one-vs-one* approach. Required parameters as *C, gamma, nu, degree* (see Chapter 4), were optimized for the best performances achieved on a grid of predefined value ranges and minimizing each time the resulting test error.
All experiments was realized considering cross-validated results. Depending on the dataset size to have an adequate number of examples per set, we divided it in 5 or 7 partitions. Circularly one of these part was used as test set and all the remaining was employed for training. Reported performance are given taking an average of these single executions.
To better test the employed feature sets we executed a high number of classifications, varying both the SVM parameters and the inputs (feature vectors). On the other hand to better catch the classification performance in underwater environment we tested the three feature sets by considering for each input image patch: 1) a variable or fixed patch size, 2) all the single RGB channel plus the gray-level one.
The first of the two points is related to evaluate performance by comparing the two way

in which we might want to process an entire image. We are interested in developing a framework that can classify each fixed-size patch (window) of an image or that can label every region of an automatic pre-segmented image. Using features from different patch size is not trivial and might point out interesting differences between the considered feature sets.
For what concern the colour differences, all the descriptors are based on single channel image. Besides getting the common gray-level analysis we want also evaluate performance on the three specific single RGB channel. Differently from the terrestrial one, the underwater environment is strongly affected by colour distortions due to the medium transmission properties (see chapter 1). We want to study the behaviour and the implications of using one colour channel instead another in different underwater scenarios. For example the red channel might be extremely informative but it is not always sufficiently present in an underwater image and so might be useless. Another of our objective is to test if and in which conditions this choice may be useful. This analysis of single channels can lead to construct an underwater classification algorithm that can take advantage of using and combine appropriately the multi-channel information.
Other than colours and single datasets, another parameter that we analyze is the choice of the appropriate SVM kernel in relation to the number of classes to be discriminated or the feature set employed.

5.1 Obtained results

It was not an easy job to synthesize all results obtained from our experiments, due to their amount and their analytical possibilities.
The focus here is only on the significant aspects of collected data. The principal measure that we use for an overall evaluation and comparison is the *accuracy* measure of performance—and underlining that a balanced number of experiments per class was kept.
In the first part we treat as separate the various configuration used for experiments. In particular we divide the two cases that employ a *variable* versus a *fixed* size patch as inputs. It has to be noted that the fixed window case presents also results on three more datasets (12 instead of 9 for the variable-size case).
Dealing with single experiments individually is not feasible here, so the analysis will be carried out showing aggregated results regarding all the different configurations. In particular we focus in how performance varies with respect to the:

- feature sets

- colour channels
- SVM kernels
- datasets
- number of classes per dataset.

5.1.1 Variable window: Feature sets

The three tested feature sets were *first order statistics* (f), *second order statistics* (s) and *uniform LBP* (l); the last one is used in $LBP^{u2}_{8,1}$ configuration (see section 4.2.3). Obtained overall performances are reported in Table 5.1. Accuracy values reported are

Table 5.1: Overall accuracy by features (mean and standard deviation) - [v]

feature set	Accuracy	(std dev)
f	52,90	24,22
l	70,52	24,12
s	44,56	18,04

averaged over all other configuration parameters (channels, datasets, kernels, etc..), and this may also explain the high standard deviation.
The l feature set appears to sharply overcome the other two. This may be also evidenced by the the chart in Figure 5.1 where is clearly visible how l features outperform taking into consideration both mean value and their standard deviation. Obviously this does not means that l features will perform better in all single experiments independently from the actual configuration. What is possible to say for now is that using LBP achieves in an overall view better results than using more statistical-based features.

5.1.2 Variable window: Colour channels

Table 5.2 reports the accuracy measured for different colour channels.

Table 5.2: Overall accuracy performance by colour channels (mean and standard deviation) - [v]

Channels	**Accuracy**	**(std dev)**
blue	59,54	25,90
gray	62,11	26,28
green	59,67	25,74
red	58,15	23,09

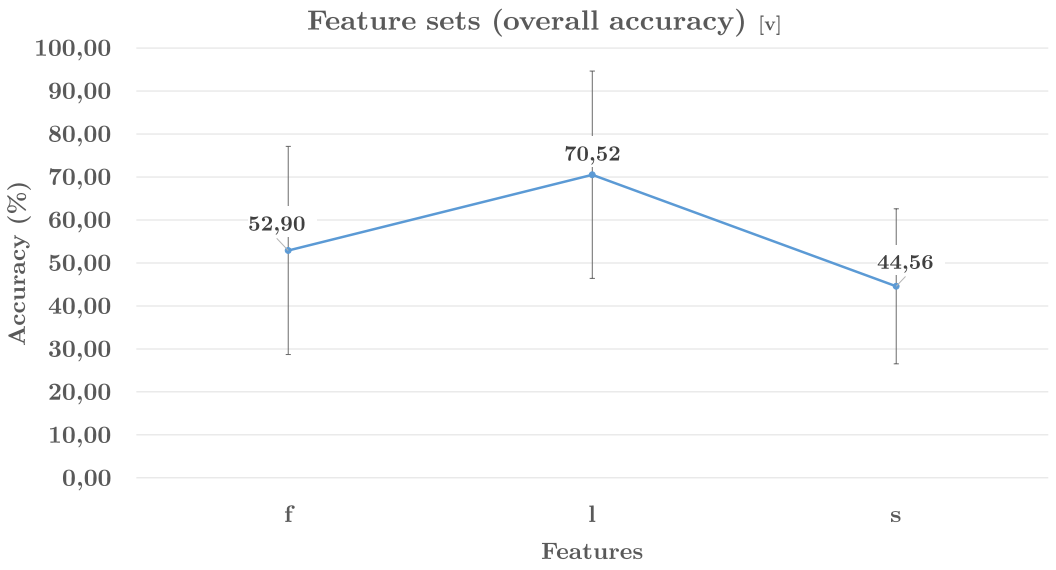


FIGURE 5.1: Overall accuracy performance by feature sets.

Results are very close to each other both considering their average accuracy and the standard deviation. Slight better performance are achieved using the gray-level image inputs while the lowest ones are obtained by using the red channel. This fact was doubtless expected keeping in mind the characteristics of underwater environment for what regards the transmission properties.
The chart in Figure 5.2 remarks how *a priori* is not possible in this case to decide which is the best colour choice and a specific evaluation of the actual dataset might be necessary.

5.1.3 Variable window: Datasets and number of classes

Tests have been conducted on nine out of twelve dataset, in particular the D1-D9 (please refer to the section 4.3 for details about these dataset, their composition and classes). Table 5.3 reports the overall accuracy performance achieved for all datasets. Obtained results largely vary both in mean value and standard deviation as can be immediately seen in chart in Figure 5.3. Anyhow we did not a priori expect an homogeneous behaviour, because every dataset has its own particularities and acquired conditions.
We notice that those datasets that achieve lower (on average) results, are those with the highest number of classes. D1 and D8 have, respectively, five and four classes while dataset D4, with only two classes, is the one that achieves best performance.
Nevertheless is not solely the number of classes that counts. In fact dataset D2 that

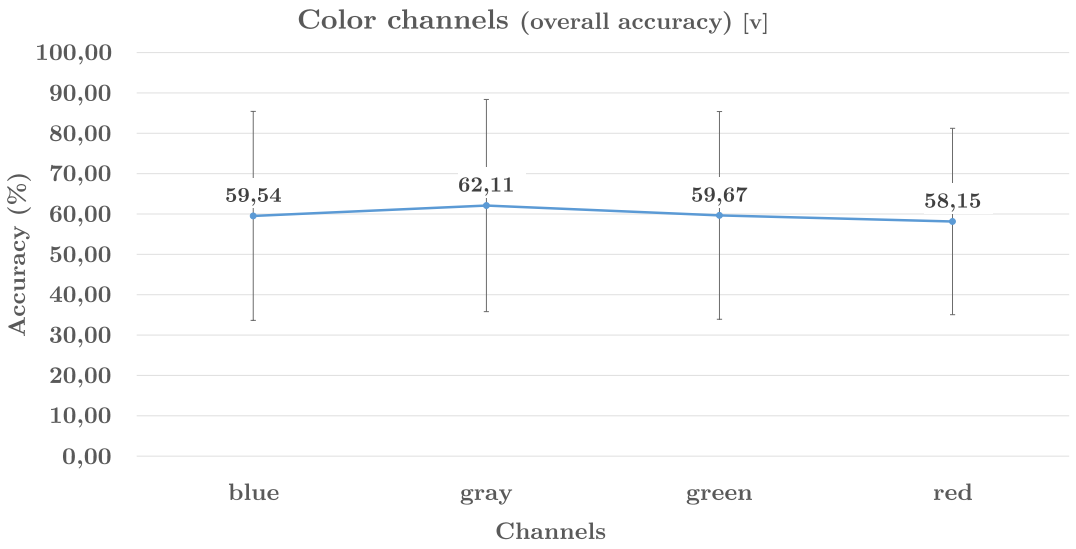


FIGURE 5.2: Overall accuracy performance by channels.

TABLE 5.3: Overall accuracy performance by datasets (mean and standard deviation) - [v]

Datasets	**Accuracy**	**(std dev)**
D1	46,13	17,70
D2	59,35	27,69
D3	62,81	22,46
D4	76,99	19,86
D5	65,10	18,45
D6	56,47	27,08
D7	57,11	33,29
D8	50,73	22,20
D9	64,13	22,82

like D1 has five classes shows significantly better performance. This confirm that overall results are widely dependent on the particular considered dataset and its intrinsic characteristics. Intra-class variations doubtless play an important role.
Table 5.4 and the relative chart in Figure 5.4 report the overall results obtained varying only the number of classes over all the experiments.

It can be noticed how average values are globally decreasing when the number of classes increase. Nevertheless, the large standard deviation values—higher for biggest numbers of classes— means that using appropriate configurations the performance can be in any case significantly improved.

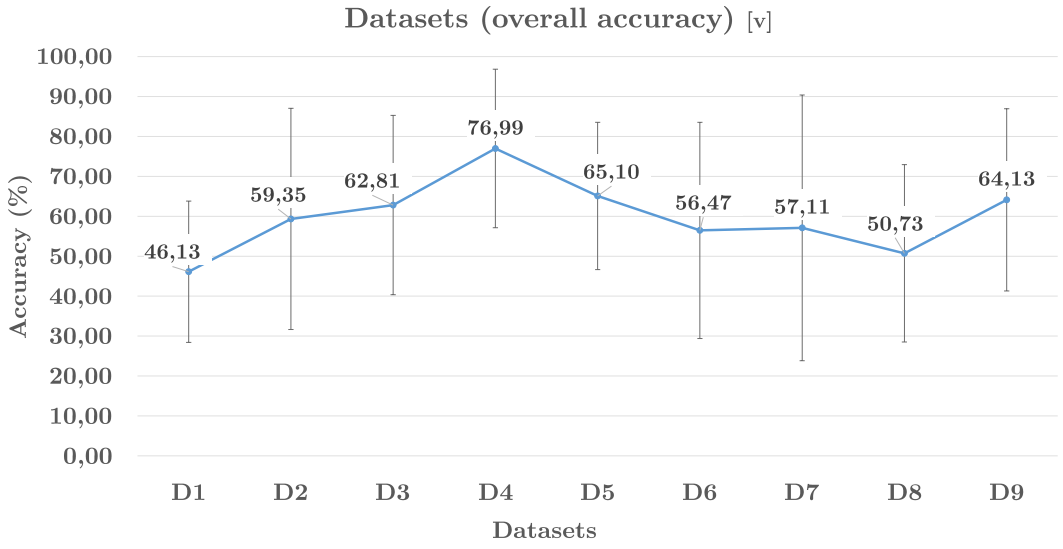


FIGURE 5.3: Overall accuracy performance by datasets D1-D9.

TABLE 5.4: Overall accuracy performance by number of classes (mean and standard deviation) - [v]

No. of classes	Accuracy	(std dev)
2	76,99	19,86
3	64,01	21,25
4	54,77	27,90
5	52,74	24,08

5.1.4 Variable window: SVM kernels

Performance over all the four kernel possibilities, *linear*, *polynomial*, *RBF* and *sigmoid*— each SVM configuration was trained by their own optimally parameters— are reported on Table 5.5 and its relative chart in Figure 5.5. It may be observed that polynomial and RBF kernels are those that realize best performance.

TABLE 5.5: Overall accuracy performance by kernels (mean and standard deviation) - [v]

Kernel	Accuracy	(std dev)
linear	57,05	28,27
poly	72,87	19,17
RBF	71,57	20,91
sigmd	37,97	12,74

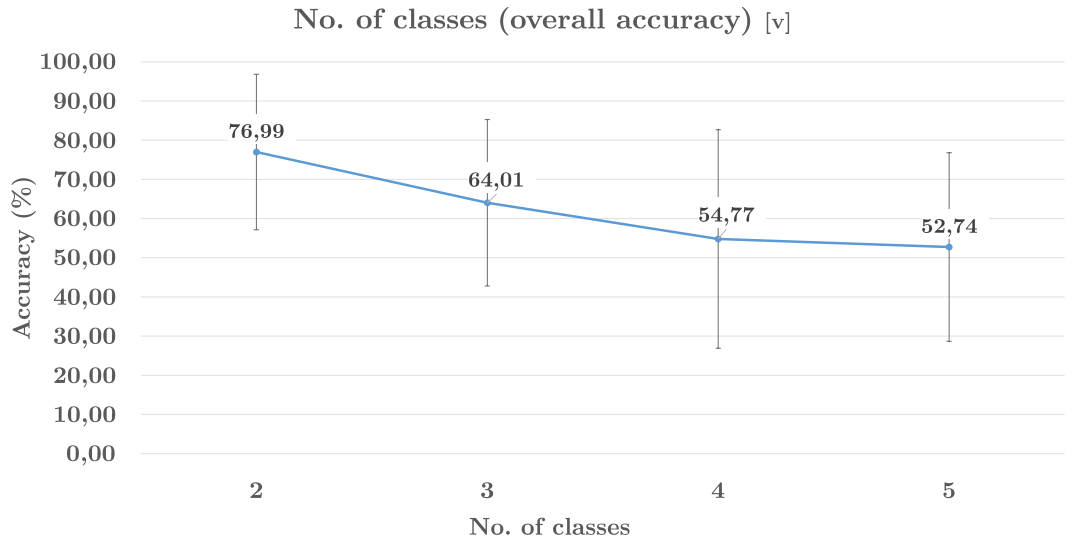


FIGURE 5.4: Overall accuracy performance by number of classes.

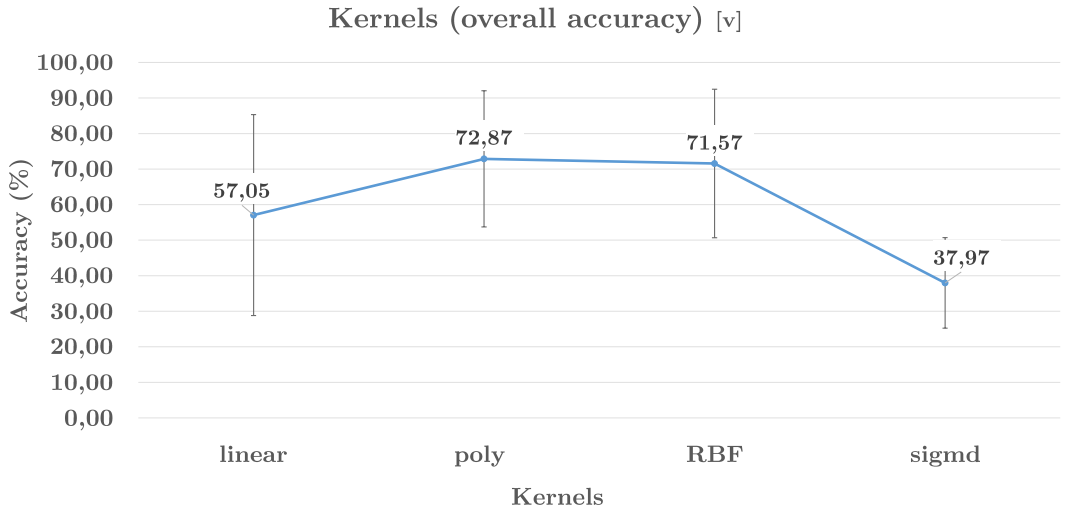


FIGURE 5.5: Overall accuracy performance by kernels.

For both, accuracy results are very close so we cannot say which of two has in general to be preferred. Looking at the standard deviations there is a close behaviour too, so we need to go in deep with the analysis of other parameters that characterize each single experiment, to appreciate substantial differences.

5.1.5 Variable window: Feature sets and channels

Table 5.6 reports performance measured by colour channels while the feature set is varied. As expected—see previous single analysis by features and by channel—this

TABLE 5.6: Overall accuracy performance by features and colour channels (mean and standard deviation) - [v]

Channels by feature sets	Accuracy	(std dev)
f	**52,90**	
blue	47,10	22,32
gray	56,73	26,73
green	48,30	22,65
red	59,45	23,55
l	**70,52**	
blue	72,09	24,53
gray	72,69	24,77
green	71,96	24,19
red	65,32	23,20
s	**44,56**	
blue	45,22	17,49
gray	45,20	18,90
green	45,12	17,69
red	42,70	18,67

Table point out the fact that the LBP-based feature set performs better regardless the colour channel. Visualizing these results (chart in Figure 5.6) we may see that the l features show an overall uniformity about the green, blue and gray channels, while red one is those characterized by the lowest performance.

On the other hand the f features seems perform slightly better with the red channel despite its lower reliability in underwater environment. Even from the chart in Figure 5.6 it is possible to note how both the two statistical approaches (f and s) give the same results on green and blue channel, the predominant colours in many underwater scenarios.

Looking at the gray-channel—that in practice is a weighted mean on the RGB channels—we may conclude that the better performance of f features than s are actually related to the red channel influence. So the red channel can be used only with features that does not encode any structural or relative information by pixel values. Otherwise, the red channel instability seems degrade the discriminative performance as in the case of l and s feature sets.

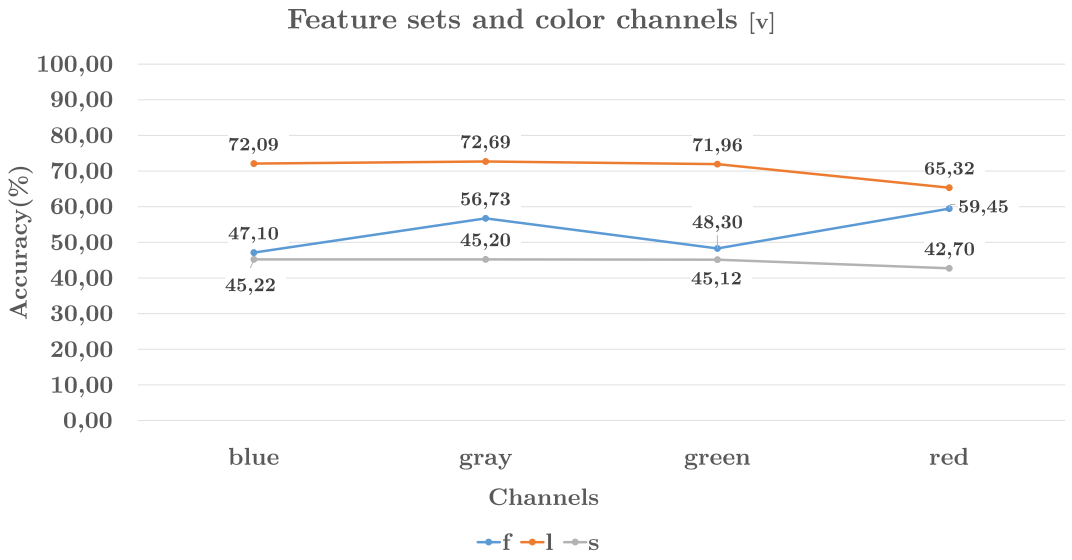


FIGURE 5.6: Overall feature set accuracy performance in relation to colour channels.

5.1.6 Variable window: Feature sets and kernels

Table 5.7 reports accuracy values and their standard deviations achieved considering the three feature sets with respect to the four kernels used for SVM classification. These

TABLE 5.7: Overall accuracy performance by feature sets and kernels (mean and standard deviation) - [v]

Kernels by feature sets	**Accuracy**	**(std dev)**
f	**52,90**	
linear	33,77	16,19
poly	57,02	23,07
RBF	79,17	14,48
sigmd	41,62	12,56
l	**70,51**	
linear	81,60	12,46
poly	83,42	12,23
RBF	82,20	12,14
sigmd	34,84	13,30
s	**44,56**	
linear	31,14	14,52
poly	66,66	12,19
RBF	41,95	9,77
sigmd	38,49	11,90

data are also visualized in chart in Figure 5.7.

The obtained accuracy values point out interesting observations.

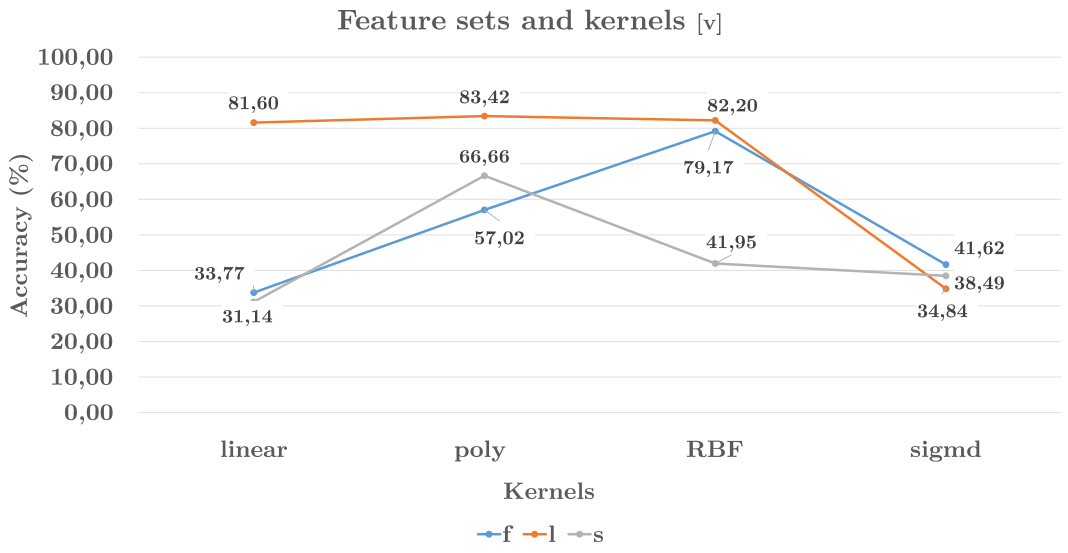


FIGURE 5.7: Overall feature set accuracy performance in relation to kernels.

First of all using the LBP-based feature set (l) achieve the best performance unless we use the sigmoid kernel—which is, by the way, the one that in general achieves worst performance with all features.
There is a little bit supremacy of the polynomial kernel over the RBF, but we think it is negligible and partially dependent on particular datasets.
First order (f) features seem to perform better only with the RBF kernel while for the second order one (s) the evidence points out that the polynomial kernel is to be preferred.
The l features may perform well also with linear kernel. This might suggest that the LBP features are able to better catch the intrinsic discriminative properties of the underwater scenario. In fact the feature vector does not necessary need to be remapped in a more complex feature space to achieve discrimination between classes. Note that the accuracy gained by using polynomial versus RBF kernel is less than 2%.

5.1.7 Variable window: Feature sets and datasets

Table 5.8 reports how the feature performance vary in relation to all the 9 dataset here considered. The l features performs again better for each single dataset. In particular all the three datasets show coherent results. If we qualitative compare the three lines of the chart in Figure 5.8 we can observe that the shapes is practically the same. With the exception of D2—where the l features goes sensibly better—all the features maintain almost constant their relative performance (i.e. a dataset that is "difficult" for one

Table 5.8: Overall accuracy performance by feature sets and datasets (mean and standard deviation) - [v]

Datasets by feature sets	**Accuracy**	**(std dev)**
f	**52,90**	
D1	39,50	16,14
D2	45,66	17,17
D3	53,32	18,34
D4	73,85	22,73
D5	59,41	25,46
D6	48,08	23,80
D7	51,13	34,09
D8	42,57	17,87
D9	62,55	22,02
l	**70,52**	
D1	53,21	14,45
D2	77,93	27,09
D3	74,68	18,78
D4	82,92	20,01
D5	73,34	15,40
D6	69,00	25,56
D7	67,25	36,79
D8	61,08	22,62
D9	75,23	19,78
s	**44,56**	
D1	35,79	17,27
D2	38,05	13,47
D3	45,28	13,52
D4	68,23	17,84
D5	56,86	10,21
D6	37,49	17,52
D7	39,85	18,99
D8	36,25	12,00
D9	43,25	12,88

feature set has the same difficulty for the other two). This means that variations in accuracy over D1-D9 datasets are strongly dependent on their actual nature.

Concerning the number of classes (the other interesting parameter to analyse) Table 5.9 and the associated chart in Figure 5.9 show—without surprise at this point—that l features continue to perform constantly better and are practically insensitive with respect to an increasing number of classes in the dataset.

Classification accuracy decrease as the number of classes increase in the same manner that has been found in chart in Figure 5.4 relative to all the three features sets together. LBP-based features have a smaller spread passing from 2 to a 5-class dataset. Accuracy values are averaged and the relative standard deviation—substantially high also in the

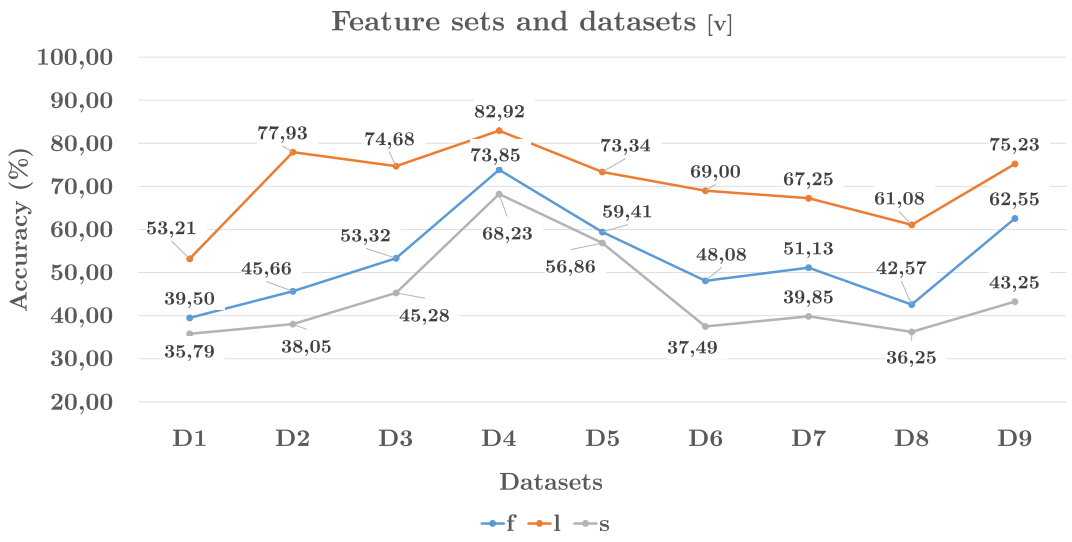


FIGURE 5.8: Overall feature set accuracy performance in relation to the D1-D9 datasets.

TABLE 5.9: Overall accuracy performance by feature sets and number of classes (mean and standard deviation) - [v]

No. of classes by feat. set	Accuracy	(std dev)
f	**52,90**	
2	73,85	22,73
3	58,43	22,00
4	47,26	25,82
5	42,58	16,68
l	**70,52**	
2	82,92	20,01
3	74,42	17,71
4	65,78	28,56
5	65,57	24,78
s	**44,56**	
2	68,23	17,84
3	48,46	13,46
4	37,86	16,16
5	36,92	15,28

case of four an five classes—may lead to argue that taking a single experiment the obtained performance can be considerably better. In particular l features seems able to achieve more than 90% of accuracy in all cases despite the number of classes.

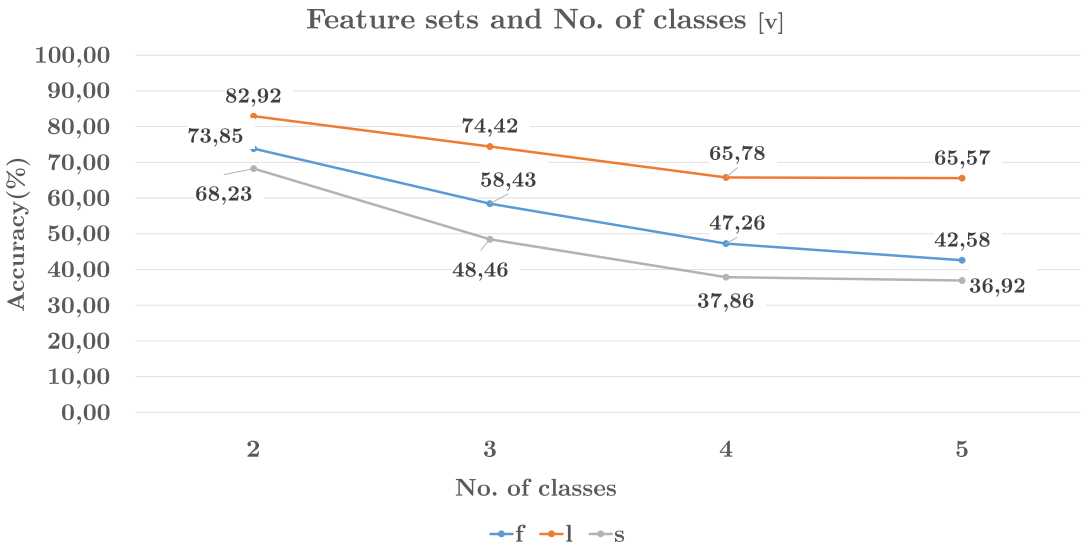


FIGURE 5.9: Overall feature set accuracy performance in relation to the number of classes in a dataset.

5.1.8 Variable window: Channels and datasets

To make our consideration about obtained results stronger we need to see if there is any correlation regarding the considered input image channel and single datasets. Theoretically, using different input colour channels might lead to have substantially different performance depending on dataset. This fact is linked to the consideration about the appearance of underwater environments (see 1.1). The possibility to achieve a good classifier that can equally well performs in all scenarios is one of the main investigations of this work.

Table 5.10 reports the accuracy (and its relative standard deviation) of different input image channels while datasets vary.

Due to the amount of data it is convenient here to have a qualitative analysis, as reported in chart in Figure 5.10. It points out a behaviour that is quite uniform across datasets. The gray-level is the one that everywhere perform slightly better. In particular datasets D6-D7-D8 and D9 show a higher spread.

Blue and green channels performs with an identical fashion. The red channel, instead, seems achieve comparable performance but not in all datasets; performance on D6 and D8 are significantly lower. Interesting is the fact that both dataset came from the same environment but are recorded with different cameras (see Figure 4.16 and 4.18 from section 4.3), so performance appear connected more to the environment itself than their acquiring modes.

The variation analysis of considered channel and the number of classes (Table 5.11 and

Table 5.10: Overall accuracy performance by colour channels and datasets (mean and standard deviation) - [v]

Datasets by channels	**Accuracy**	**(std dev)**
blue	**59,54**	
D1	45,69	18,49
D2	58,90	29,25
D3	62,69	23,13
D4	76,07	20,05
D5	65,37	19,93
D6	58,40	28,83
D7	54,79	34,24
D8	51,47	24,64
D9	62,48	24,63
gray	**62,11**	
D1	48,70	18,99
D2	60,86	29,77
D3	62,84	23,56
D4	77,45	23,75
D5	66,53	17,10
D6	60,20	30,48
D7	59,05	35,87
D8	55,43	24,22
D9	67,93	23,68
green	**59,67**	
D1	45,51	19,18
D2	57,25	28,35
D3	62,87	22,79
D4	76,40	19,63
D5	66,78	17,87
D6	57,79	29,55
D7	54,98	34,49
D8	51,91	24,49
D9	63,58	23,83
red	**58,15**	
D1	44,60	15,33
D2	60,38	25,79
D3	62,84	22,56
D4	78,05	17,40
D5	61,73	20,09
D6	49,48	19,34
D7	59,62	31,38
D8	44,12	14,30
D9	62,53	20,76

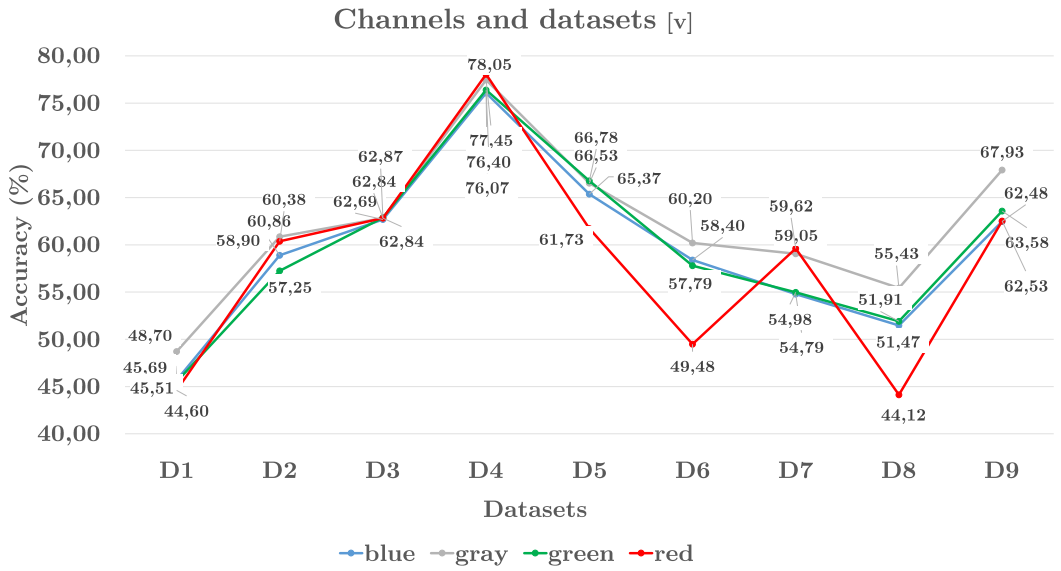


FIGURE 5.10: Overall accuracy performance of single image channels varying the datasets.

chart in Figure 5.11) does not point out important new considerations and confirms the previous one obtained. Gray-level seems again a little bit better than the other channels, considering both the mean accuracy and its standard deviation, in practically all cases.

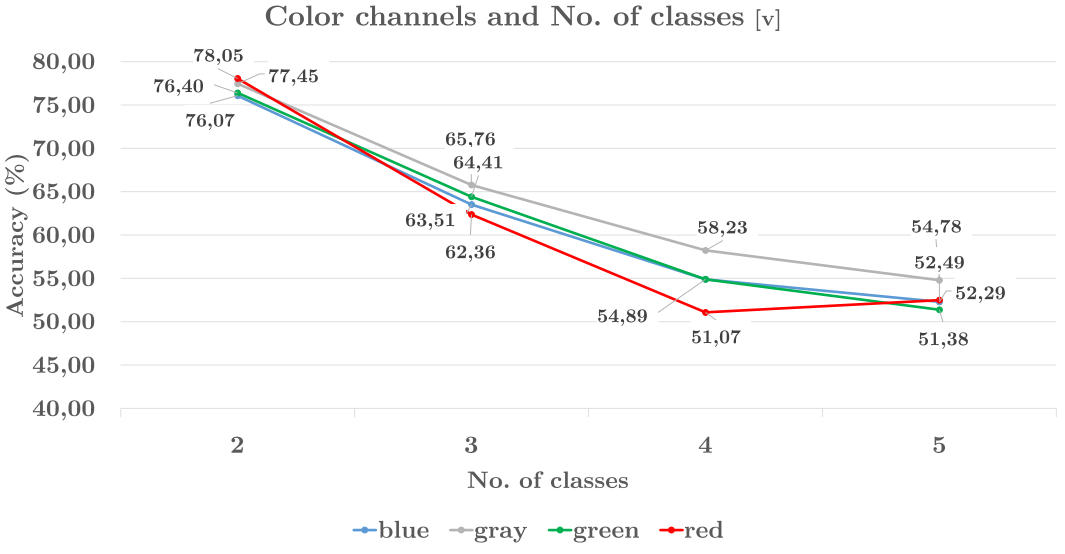


FIGURE 5.11: Overall accuracy performance of single image channels by varying the number of classes.

TABLE 5.11: Overall accuracy performance by colour channels in relation to the number of classes (mean and standard deviation) - [v]

No. of classes by channels	Accuracy	(std dev)
blue	**59,54**	
2	76,07	20,05
3	63,51	22,20
4	54,89	29,00
5	52,29	24,98
gray	**62,11**	
2	77,45	23,75
3	65,76	21,31
4	58,23	29,97
5	54,78	25,33
green	**59,67**	
2	76,40	19,63
3	64,41	21,26
4	54,89	29,25
5	51,38	24,54
red	**58,15**	
2	78,05	17,40
3	62,36	20,71
4	51,07	23,26
5	52,49	22,35

5.1.9 Variable window: Kernels and datasets

Table 5.12 and its related chart in Figure 5.12, report the accuracy (mean and standard deviation) of using the four kernels in every dataset D1-D9.

Here it is possible to note a global coherent behaviour—i.e. there is not evidence that some kernel performs better over a single dataset. In particular, as for previous conducted analysis about SVM configurations, the polynomial kernel achieves slight better performance as can be seen also in Table 5.13, where the mean accuracy obtained with polynomial kernel is compared (with a simple subtraction) with all other kernels. Linear and sigmoid kernels are completely dominated in all datasets—with a significant spread— while the RBF one is only just a little overcome.

Table 5.14 and chart in Figure 5.13 report, the behaviour of kernel in relation to the number of classes per dataset. Conclusions are the same and there are not evident correlations in this sense. Globally there are shared descending lines but all of them have practically the same behaviour—except for the sigmoid kernel in the 4-class datasets—with the increase of dataset classes.

Table 5.12: Overall accuracy performance by kernel and dataset (mean and standard deviation) - [v]

Datasets by kernel	Accuracy	(std dev)
linear		
D1	40,35	23,28
D2	58,08	34,34
D3	59,72	28,74
D4	74,90	22,27
D5	59,08	19,73
D6	53,08	33,08
D7	56,13	33,14
D8	52,18	23,95
D9	59,95	26,83
poly		
D1	58,05	11,12
D2	71,89	22,09
D3	74,83	15,73
D4	90,44	11,23
D5	76,88	12,91
D6	70,67	19,94
D7	75,22	22,86
D8	60,96	17,82
D9	76,93	18,63
RBF		
D1	54,11	13,14
D2	72,15	23,25
D3	73,41	18,80
D4	84,97	18,06
D5	76,13	16,04
D6	68,92	24,52
D7	78,26	19,55
D8	61,18	20,45
D9	75,03	19,77
sigmd		
D1	31,99	2,85
D2	35,28	7,00
D3	43,26	0,07
D4	57,66	6,69
D5	48,32	0,23
D6	33,20	4,10
D7	18,82	16,11
D8	28,60	2,67
D9	44,60	1,74

TABLE 5.13: Accuracy spread in relation to the polynomial kernel - [v]

Dataset	(poly-RBF)	(poly-linear)	(poly-sigmd)
D1	3,94	17,70	26,06
D2	-0,26	13,81	36,61
D3	1,42	15,11	31,57
D4	5,47	15,53	32,77
D5	0,75	17,80	28,56
D6	1,75	17,59	37,47
D7	-3,04	19,09	56,40
D8	3,94	8,78	32,36
D9	1,90	16,98	32,33
Mean	**1,76**	**15,82**	**34,90**

TABLE 5.14: Overall accuracy performance by kernel and number of classes (mean and standard deviation) - [v]

No. of classes by kernel	Accuracy	(std dev)
linear	**57,05**	
2	74,90	22,27
3	59,58	24,85
4	53,80	29,76
5	49,21	30,23
poly	**72,87**	
2	90,44	11,23
3	76,21	15,62
4	68,95	20,76
5	64,97	18,58
RBF	**71,57**	
2	84,97	18,06
3	74,86	17,91
4	69,45	22,30
5	63,13	20,72
sigmd	**37,97**	
2	57,66	6,69
3	45,39	2,38
4	26,87	11,28
5	33,63	5,52

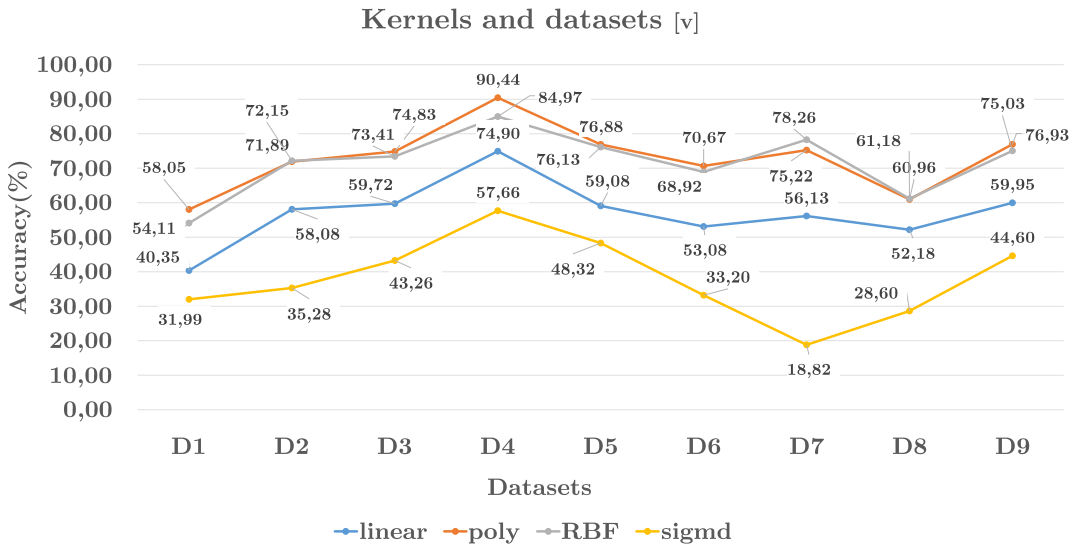


FIGURE 5.12: Overall kernel accuracy performance by datasets D1-D9.

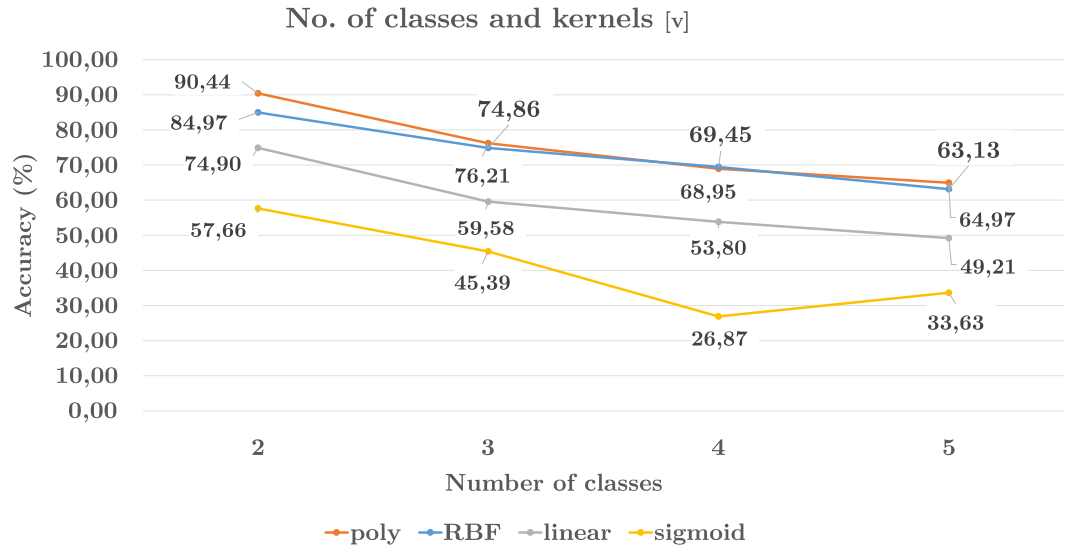


FIGURE 5.13: Overall kernel accuracy performance by the number of classes.

5.2 Fixed window

In the following sections an analysis similar to that achieved for variable-size patches is conducted.

Now the focus is about considering *fixed-size* input image patches. We notice that fixed windows may be extracted from images when a pre-segmentation over images cannot be performed. In fact, in certain conditions, the time required for an image preprocessing

may be unacceptable.
Our objective will be then to compare both results, obtained from fixed and variable patch size and see which are the appreciable advantages/disadvantages in using one approach instead of another.
For evaluating fixed image patches we also have the availability of three more datasets, D10-D12.

5.2.1 Fixed window: Features

The first analysis that we briefly report, is related to the overall accuracy performance on the three employed feature sets: *first order statistics* (f), *second order statistics* (s) and *uniform LBP pattern* (l) in $LBP^{u2}_{8,1}$ configuration.
Table 5.15 reports the obtained performance averaged over all the other parameters as kernel, datasets and colour channels.
Results are clear enough. LBP-based feature set (l) achieve the better performance with

TABLE 5.15: Overall accuracy by feature set (mean and standard deviation) - [f]

feature set	Accuracy	(std dev)
f	53,95	22,74
l	74,58	21,32
s	46,90	19,39

more than 20% difference with the second best feature set (first order statistics, f). The chart in Figure 5.14 visualize this spread that appears consistent also considering the relative standard deviations.
The l features appears doubtless the better choice for this classification task.

5.2.2 Fixed window: Colour channels

Table 5.16 and its relative chart in Figure 5.15 show the performance obtained by varying the input colour channel. With the exception of the gray-level channel—that achieve better results—there is not at this level a relevant distinction about the other three channel, red, green and blue.

In particular, blue and green carry out approximatively identical performance while the red channel results again the worst to be used for classify the input image patches.

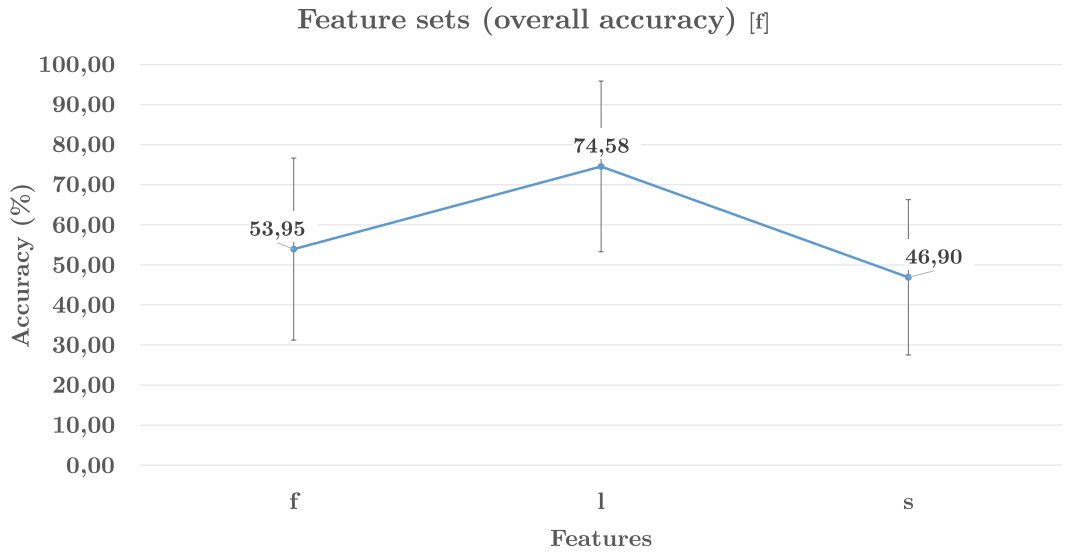


FIGURE 5.14: Accuracy performance of the three different feature sets over all experiments.

TABLE 5.16: Overall accuracy performance by colour channels (mean and standard deviation) - [f]

Channels	Accuracy	(std dev)
blue	62,18	25,18
gray	64,92	24,60
green	62,57	24,87
red	59,72	22,90

5.2.3 Fixed window: Datasets and number of classes

In comparison to the case with variable-size patches, now the available dataset to conduct experiments are three more, so in total we have 12 datasets (for more details see 4.3). Table 5.17 reports the average accuracy performance obtained for every dataset. In particular the chart in Figure 5.16 shows how all these data are characterized by a high standard deviation, so although all accuracy values are in the range from 50% to 80%, with appropriate configuration each dataset may achieve values largely over the 70% of accuracy.

The new dataset D10 is the one that realize the best overall performance. As we can see from Table 5.18 and chart in Figure 5.17 the mean accuracy—not surprisingly—decrease while the number of classes increases.

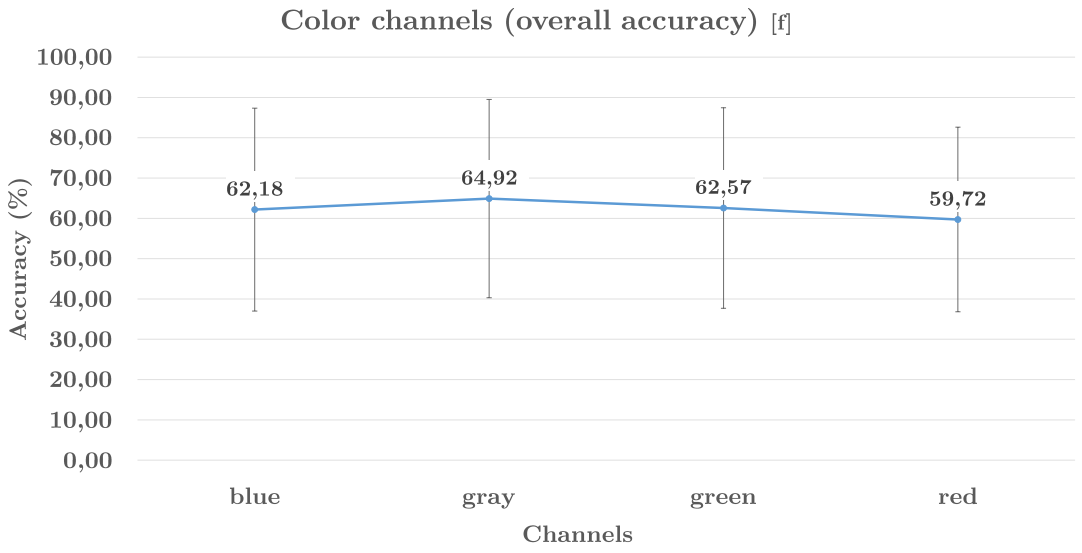


FIGURE 5.15: Overall accuracy performance by colour channels.

TABLE 5.17: Overall accuracy performance by datasets (mean and standard deviation) - [f]

Datasets	**Accuracy**	**(std dev)**
D01	54,11	27,14
D02	51,31	19,92
D03	59,11	16,19
D04	67,78	20,11
D05	66,43	17,95
D06	64,25	31,13
D07	59,20	31,15
D08	51,40	22,52
D09	64,14	24,43
D10	77,17	20,15
D11	68,20	20,83
D12	65,07	24,58

Datasets D1, D2 and D8 are those with lower values, but also with higher number of classes that may explain this behaviour. Anyhow, also D7 and D6, have the same number of classes meaning that the connection with the accuracy is not straight but is also related to other intrinsic characteristics of the dataset.

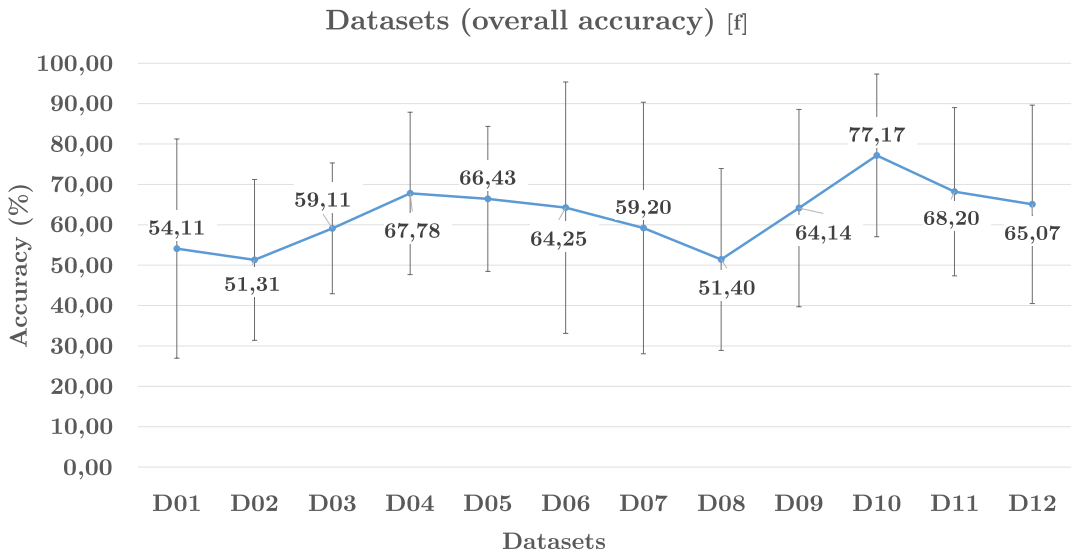


FIGURE 5.16: Overall accuracy performance by datasets.

TABLE 5.18: Overall accuracy performance by number of classes (mean and standard deviation) - [f]

No. of classes	Accuracy	(std dev)
2	72,48	20,59
3	64,59	21,16
4	58,28	28,90
5	52,71	23,75

5.2.4 Fixed window: Kernels

Concerning the employed kernels used to train the SVMs we may notice (see Table 5.19 and chart in Figure 5.18) that the RBF and polynomial kernel are again those with better mean accuracy over all experiments.

TABLE 5.19: Overall accuracy performance by kernel (mean and standard deviation) - [f]

Kernel	Accuracy	(std dev)
linear	59,31	28,40
poly	73,21	19,98
RBF	74,38	19,99
sigmd	42,49	10,98

Although the accuracy seems not so high, due to its standard deviation value, the linear kernel may be in some cases an equally good choice too.

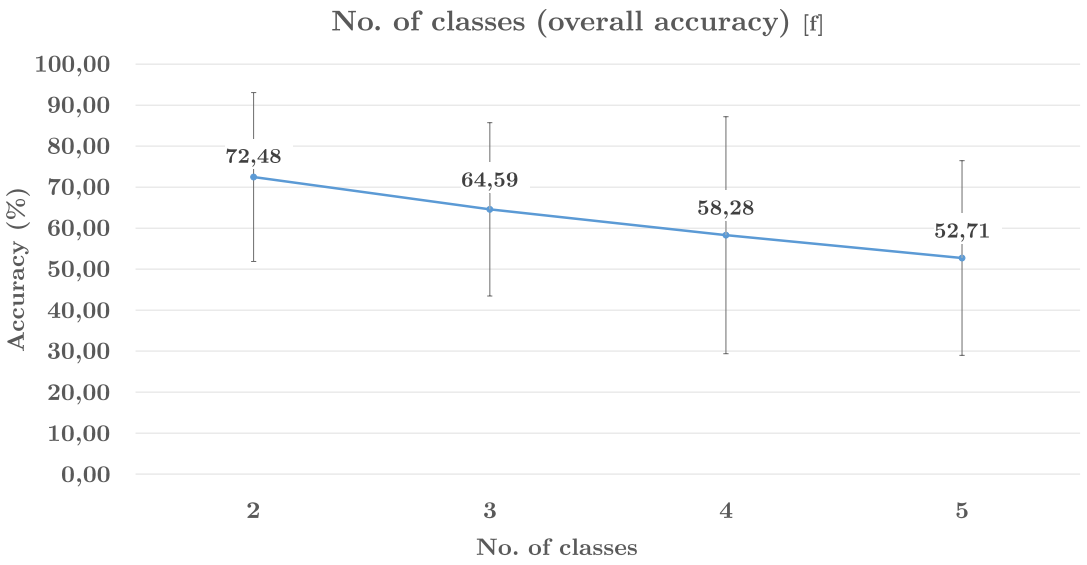


FIGURE 5.17: Overall accuracy performance by number of classes.

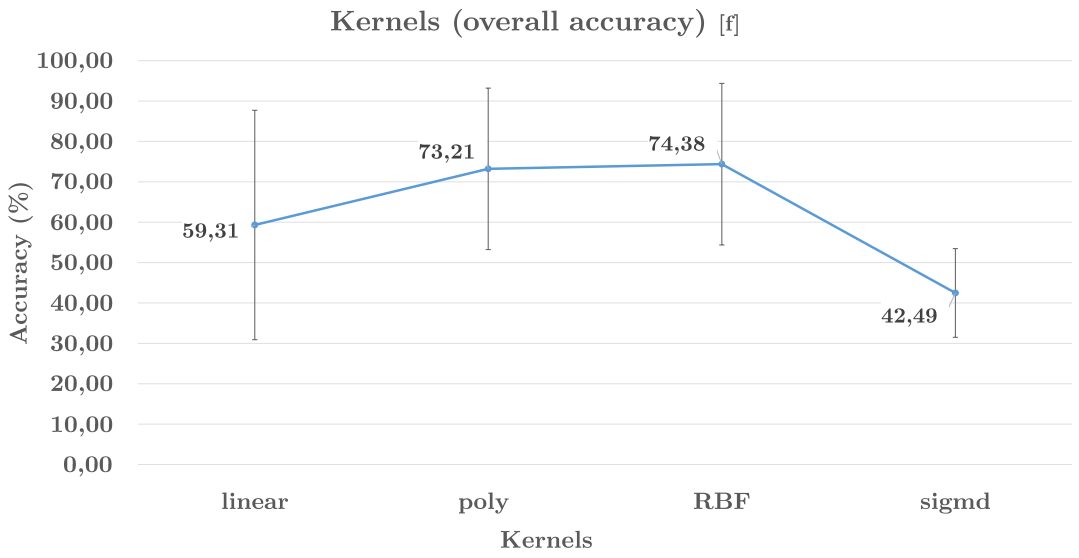


FIGURE 5.18: Overall accuracy performance by kernels.

5.2.5 Fixed window: Feature sets and channels

Table 5.20 and the related chart in Figure 5.19 investigate the possibility of a relation between the used feature set and a particular colour channel.

Data confirms what was pointed out for the analysis of features and colours in their

TABLE 5.20: Overall accuracy performance by feature sets and colour channels (mean and standard deviation) - [f]

Channels by feature sets	Accuracy	(std dev)
f	**53,95**	
blue	50,85	23,48
gray	60,76	23,85
green	52,23	22,00
red	51,96	20,83
l	**74,58**	
blue	75,34	21,93
gray	75,67	21,11
green	75,86	22,15
red	71,46	20,41
s	**46,90**	
blue	47,79	19,92
gray	47,39	20,21
green	47,42	19,51
red	45,00	18,38

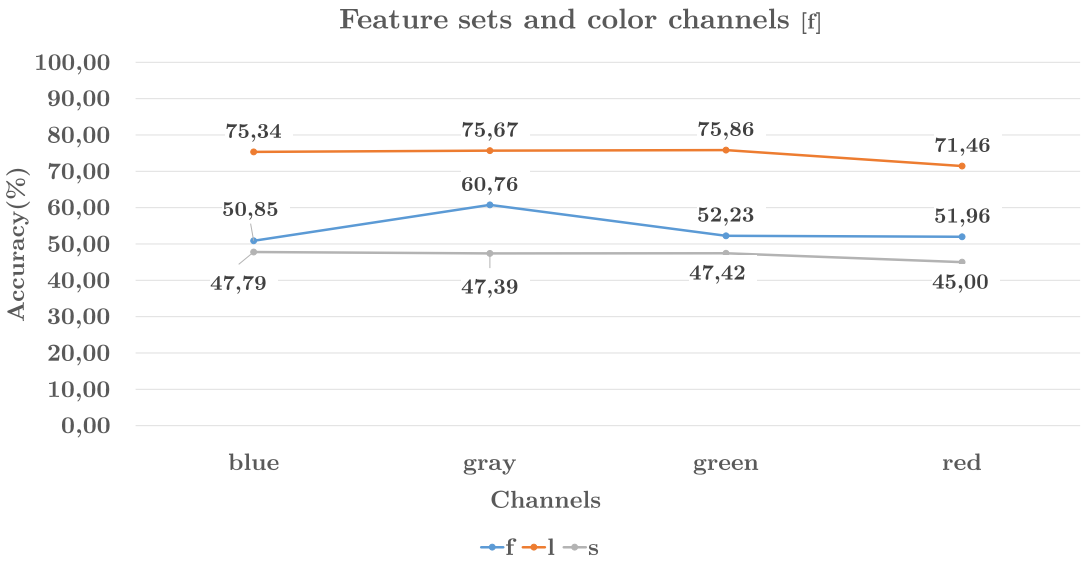


FIGURE 5.19: Accuracy performance by feature sets in combination with a particular colour channel.

standalone ways. There is no evidence of correlation between these configuration parameters.

Using the first order feature set (f) in combination with the gray-level channel seems to significantly improve performance, that remain anyhow, lower than the overall results achieved by the LBP-based features with any image colour.

5.2.6 Fixed window: Feature sets and kernels

For the analysis of the relation between feature sets and kernel, using fixed-size image as input we obtain the chart in Figure 5.20 based on data from Table 5.21. With all the

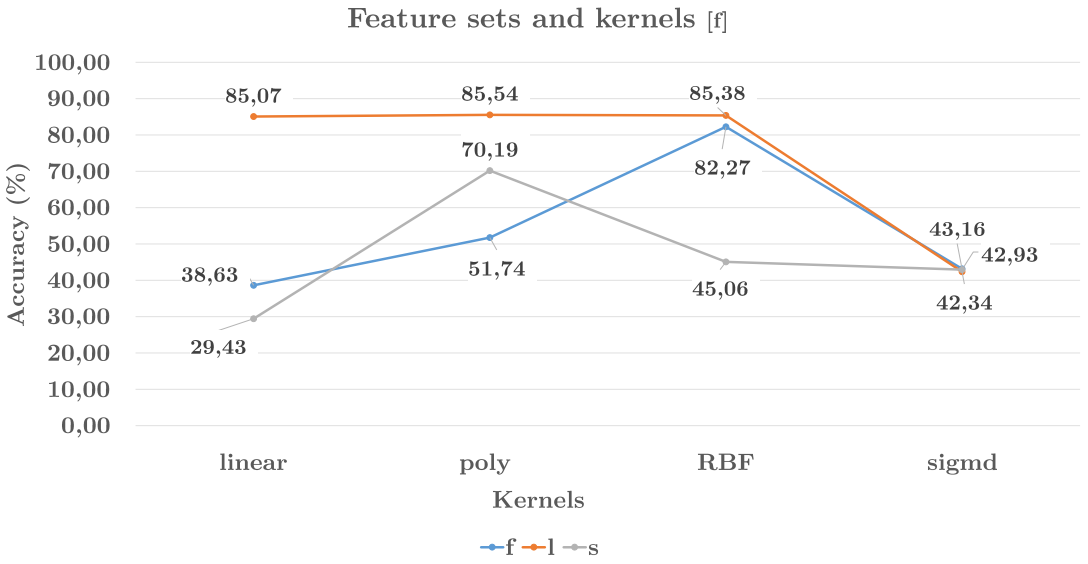


FIGURE 5.20: Accuracy performance by feature sets in combination with a particular kernel.

TABLE 5.21: Overall accuracy performance by feature sets and kernels (mean and standard deviation) - [f]

Kernels by feature set	Accuracy	(std dev)
f	**53,95**	
linear	38,63	15,58
poly	51,74	19,94
RBF	82,27	12,48
sigmd	43,16	11,10
l	**74,58**	
linear	85,07	10,66
poly	85,54	10,03
RBF	85,38	9,77
sigmd	42,34	11,01
s	**46,90**	
linear	29,43	12,60
poly	70,19	15,81
RBF	45,06	10,37
sigmd	42,93	11,27

three best performing kernels—linear, polynomial and RBF— the LBP-based features

are every time those with the best accuracy measures.
We may observe that also the linear kernel shows good results, but only when used with the RBF kernel. Despite their simplicity, first order statistics feature set seems to reach better discrimination performance when is mapped in a higher dimensional space.
LBP-based features does not seems necessary require the same treatment and just using a linear SVM the accuracy—considering the aggregated results—is practically the same than other more complex kernels.

5.2.7 Fixed window: Feature sets and datasets

Table 5.22 shows the results obtained for each dataset D1-D12 in relation to all feature sets. For a qualitative evaluation the chart in Figure 5.21 can be seen.

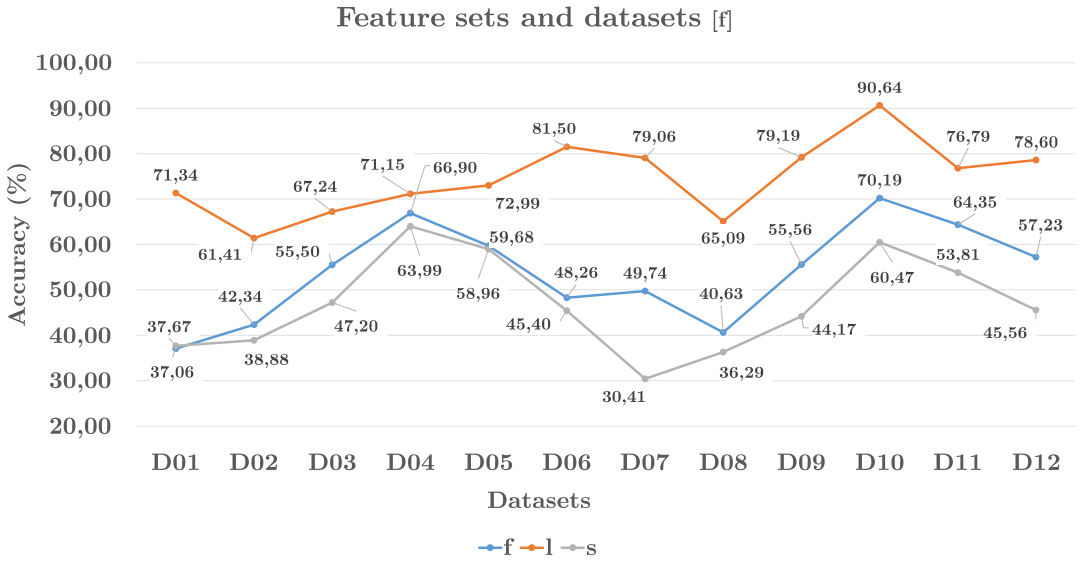


FIGURE 5.21: Accuracy performance by feature sets in every dataset.

We can observe that the best performances of LBP-based (l) feature sets are confirmed on all the twelve datasets.
In comparison to the other two feature types (f and s) that show a very close behaviour, the lines related to l features in the chart present much more uniformity.
D2 and D7 are datasets where LBP-based features have the worst performance, but anyhow the other two descriptors do the same. Dataset D6 and D7 are those with the biggest gap between the l and the other two feature sets.
On the other side, it is the dataset D2 that achieves the closest accuracy values among all descriptors. This fact might be explained with the good performance that the two statistical-based feature sets seems achieve in all the 2-class problems. In Table 5.23

Table 5.22: Overall accuracy performance by feature sets and datasets (mean and standard deviation) - [f]

Datasets by feature set	Accuracy	(std dev)
f	**53,95**	
D01	37,06	18,93
D02	42,34	17,36
D03	55,50	16,95
D04	66,90	23,34
D05	59,68	24,27
D06	48,26	27,47
D07	49,74	26,72
D08	40,63	16,60
D09	55,56	23,09
D10	70,19	11,48
D11	64,35	17,54
D12	57,23	21,68
l	**74,58**	
D01	71,34	25,58
D02	61,41	17,76
D03	67,24	12,07
D04	71,15	17,15
D05	72,99	14,21
D06	81,50	28,10
D07	79,06	26,59
D08	65,09	22,18
D09	79,19	21,92
D10	90,64	14,90
D11	76,79	18,22
D12	78,60	19,87
s	**46,90**	
D01	37,67	16,06
D02	38,88	14,25
D03	47,20	15,45
D04	63,99	20,15
D05	58,96	11,11
D06	45,40	20,68
D07	30,41	14,95
D08	36,29	12,85
D09	44,17	10,59
D10	60,47	22,79
D11	53,81	20,46
D12	45,56	19,95

and chart in Figure 5.22, for completeness are finally reported the performance of all the three feature sets while varying the number of classes instead the datasets.

TABLE 5.23: Overall accuracy performance by feature sets and number of classes (mean and standard deviation) - [f]

No. of classes by feat. set	Accuracy	(std dev)
f	**53,95**	
2	68,54	18,17
3	58,46	20,65
4	46,21	23,94
5	39,70	18,07
l	**74,58**	
2	80,90	18,65
3	74,96	17,75
4	75,22	26,23
5	66,37	22,24
s	**46,90**	
2	62,23	21,24
3	49,94	16,64
4	37,37	17,30
5	38,27	14,95

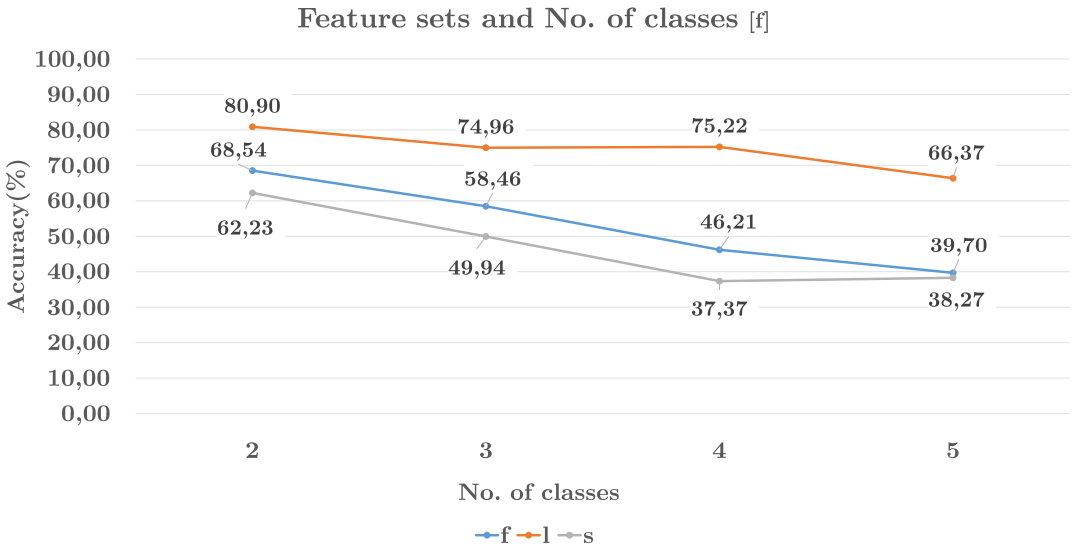


FIGURE 5.22: Accuracy performance by feature sets when is varied the number of classes.

5.2.8 Fixed window: Channels and datasets

Colour channel might be a priori connected to the dataset performance variations because the difference in datasets mostly are determined by a change in environment and/or acquisition modalities.
From the Table 5.24 and chart in Figure 5.23 we can see that actually the used image colour channel does not have a significant effect on accuracy.

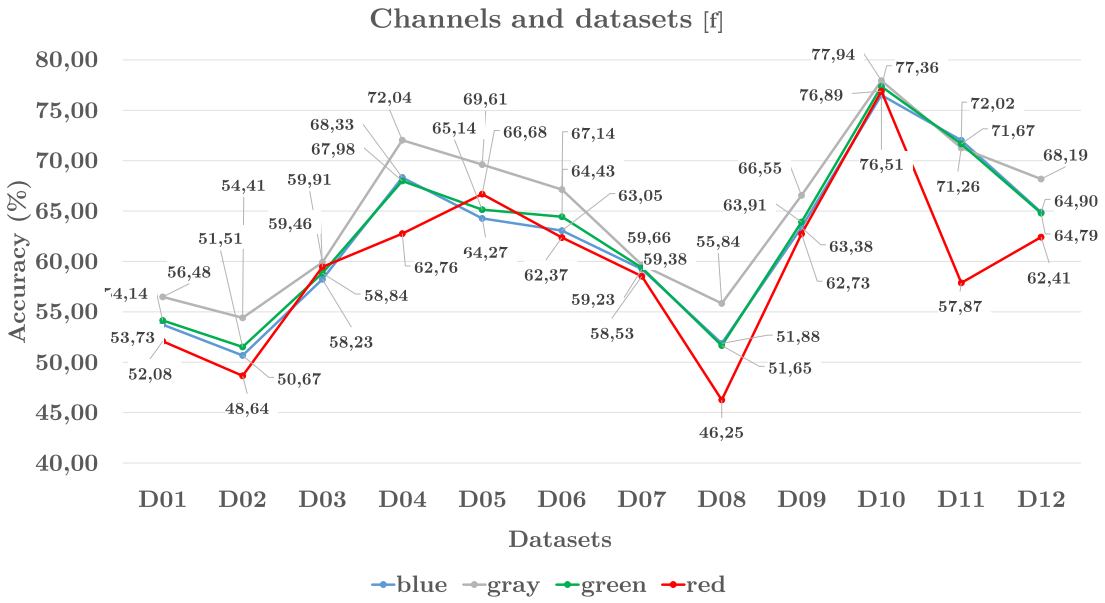


FIGURE 5.23: Accuracy performance by channels and datasets.

Clearly, in the chart some slight differences are appreciable and all seem to suggest that the gray-level channel should be the preferred choice. This gray-level predominance is much more evident if we consider as variable not the datasets but their number of classes (Table 5.25, chart in Figure 5.24).

The chart sharply underline exactly the same descending behaviour between colour channels when there is an increase of classes.
Red channel performance are constantly 5% under the gray-level line which is overall better.

5.2.9 Fixed window: Kernels and datasets

Using RBF or polynomial kernel, again, leads to the best results in term of classification accuracy as shown in chart in Figure 5.25 with data from Table 5.26 .

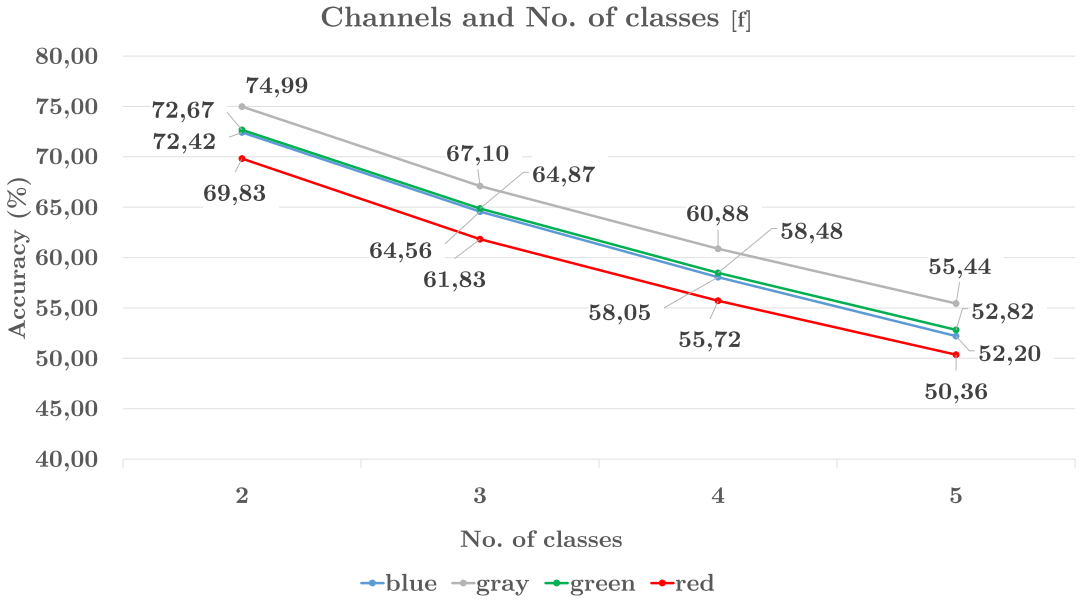


FIGURE 5.24: Accuracy performance by channels and number of classes.

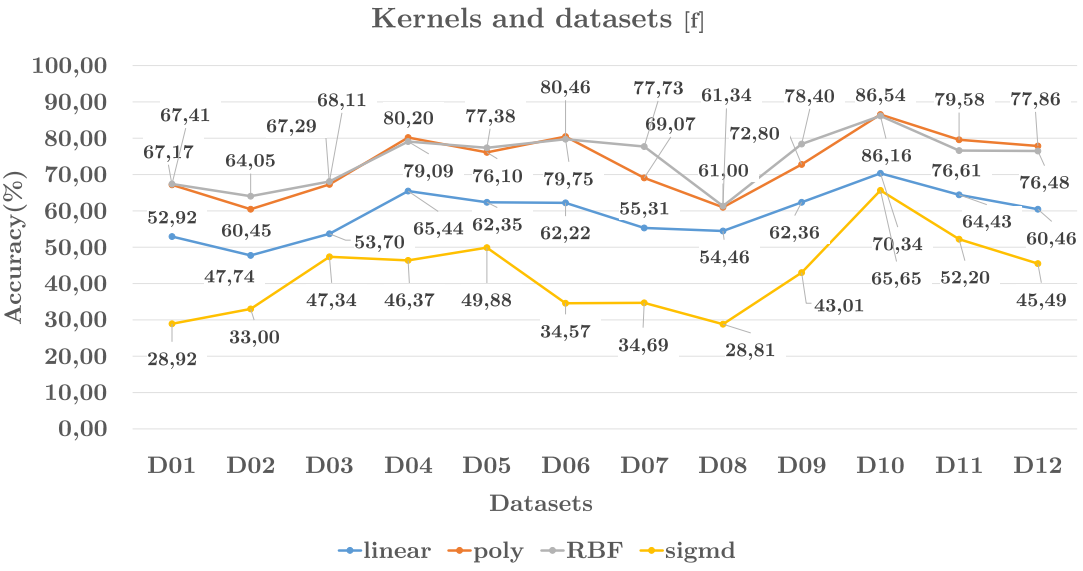


FIGURE 5.25: Accuracy performance by datasets with different SVM kernels.

The performance of both these methods are close. Only if we precisely measure the difference values in every dataset we can observe (Table 5.27) that RBF is slightly better and gains, on average, approximatively 1% in accuracy.

Actually this result is due mostly thanks to the contribution of dataset D7 and D9. In comparison to linear and sigmoid kernel the gap is, instead, higher than 10%.

Finally, considering the variations in performance on the number of classes (Table 5.28, chart in Figure 5.26), the results do not tell us nothing of different about the best

performing kernels, that remain the same. The conclusion is that, unless isolated ex-

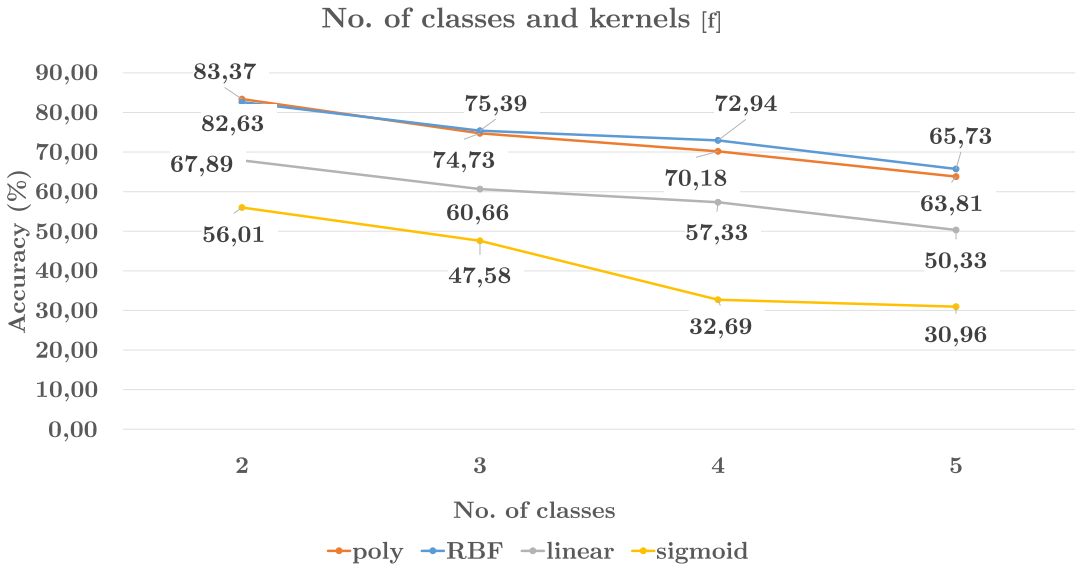


FIGURE 5.26: Accuracy performance by kernels and number of classes.

ceptional experiments and without extra knowledge about the environment, an SVM trained with RBF or Polynomial kernels should be the first choice in underwater classification.

Table 5.24: Overall accuracy performance by colour channels and datasets (mean and standard deviation) - [f]

Datasets by channel	Accuracy	(std dev)
blue	**62,18**	
D01	53,73	28,06
D02	50,67	19,97
D03	58,23	15,44
D04	68,33	19,18
D05	64,27	19,57
D06	63,05	33,30
D07	59,23	33,36
D08	51,88	23,35
D09	63,38	26,41
D10	76,51	21,31
D11	72,02	23,35
D12	64,90	26,41
gray	**64,92**	
D01	56,48	27,76
D02	54,41	21,30
D03	59,91	18,28
D04	72,04	24,29
D05	69,61	17,31
D06	67,14	30,65
D07	59,66	32,31
D08	55,84	23,14
D09	66,55	23,72
D10	77,94	21,15
D11	71,26	21,94
D12	68,19	24,26
green	**62,57**	
D01	54,14	26,84
D02	51,51	20,75
D03	58,84	15,75
D04	67,98	20,81
D05	65,14	18,31
D06	64,43	31,54
D07	59,38	33,09
D08	51,65	23,23
D09	63,91	25,50
D10	77,36	19,88
D11	71,67	23,71
D12	64,79	27,27
red	**59,72**	
D01	52,08	28,36
D02	48,64	19,13
D03	59,46	16,69
D04	62,76	16,14
D05	66,68	17,82
D06	62,37	31,81
D07	58,53	28,67
D08	46,25	21,45
D09	62,73	24,24
D10	76,89	20,16
D11	57,87	8,97
D12	62,41	22,12

TABLE 5.25: Overall accuracy performance by colour channels in relation to the number of classes (mean and standard deviation) - [f]

No. of classes by channels	**Accuracy**	**(std dev)**
blue	**62,18**	
2,00	72,42	20,37
3,00	64,56	22,50
4,00	58,05	30,08
5,00	52,20	24,01
gray	**64,92**	
2,00	74,99	22,61
3,00	67,10	21,11
4,00	60,88	28,75
5,00	55,44	24,36
green	**62,57**	
2,00	72,67	20,58
3,00	64,87	22,34
4,00	58,48	29,45
5,00	52,82	23,63
red	**59,72**	
2,00	69,83	19,35
3,00	61,83	18,50
4,00	55,72	27,94
5,00	50,36	23,86

Table 5.26: Overall accuracy performance by kernels and datasets (mean and standard deviation) - [f]

Datasets by kernel	Accuracy	(std dev)
linear	**59,31**	
D01	52,92	33,39
D02	47,74	23,92
D03	53,70	22,25
D04	65,44	16,77
D05	62,35	19,86
D06	62,22	36,66
D07	55,31	39,94
D08	54,46	23,48
D09	62,36	29,37
D10	70,34	30,85
D11	64,43	25,20
D12	60,46	31,33
poly	**73,21**	
D01	67,17	22,44
D02	60,45	15,93
D03	67,29	10,49
D04	80,20	16,48
D05	76,10	14,16
D06	80,46	24,79
D07	69,07	27,31
D08	61,00	19,31
D09	72,80	20,08
D10	86,54	13,06
D11	79,58	17,48
D12	77,86	18,24
RBF	**74,38**	
D01	67,41	20,54
D02	64,05	13,75
D03	68,11	12,62
D04	79,09	15,74
D05	77,38	14,85
D06	79,75	25,09
D07	77,73	24,63
D08	61,34	21,15
D09	78,40	21,49
D10	86,16	14,54
D11	76,61	19,42
D12	76,48	21,85
sigmd	**42,49**	
D01	28,92	1,74
D02	33,00	4,27
D03	47,34	0,49
D04	46,37	10,08
D05	49,88	0,13
D06	34,57	0,50
D07	34,69	0,26
D08	28,81	0,97
D09	43,01	1,77
D10	65,65	0,00
D11	52,20	1,45
D12	45,49	0,00

TABLE 5.27: Accuracy spread in relation to the polynomial kernel - [f]

Dataset	**(poly-RBF)**	**(poly-linear)**	**(poly-sigmd)**
D01	-0,24	14,25	20,70
D02	-3,60	12,72	11,66
D03	-0,82	13,59	10,00
D04	1,11	14,76	6,40
D05	-1,28	13,75	14,03
D06	0,71	18,24	24,29
D07	-8,65	13,76	27,05
D08	-0,34	6,53	18,34
D09	-5,60	10,44	18,31
D10	0,38	16,20	13,06
D11	2,97	15,15	16,03
D12	1,38	17,40	18,24
Mean	**-1,17**	**13,90**	**16,51**

TABLE 5.28: Overall accuracy performance by kernel and number of classes (mean and standard deviation) - [f]

No. of classes by kernel	**Accuracy**	**(std dev)**
linear	**59,31**	
2	67,89	24,55
3	60,66	25,56
4	57,33	33,56
5	50,33	28,70
poly	**73,21**	
2	83,37	14,98
3	74,73	16,61
4	70,18	24,87
5	63,81	19,45
RBF	**74,38**	
2	82,63	15,33
3	75,39	18,33
4	72,94	24,63
5	65,73	17,28
sigmd	**42,49**	
2	56,01	12,04
3	47,58	3,40
4	32,69	2,84
5	30,96	3,82

5.2.10 Fixed versus variable-size: A comparison

This section is dedicated to evaluate the measured performance in case of using variable-size versus fixed-size image patches (Figure 5.27). The reasons behind this choice may be

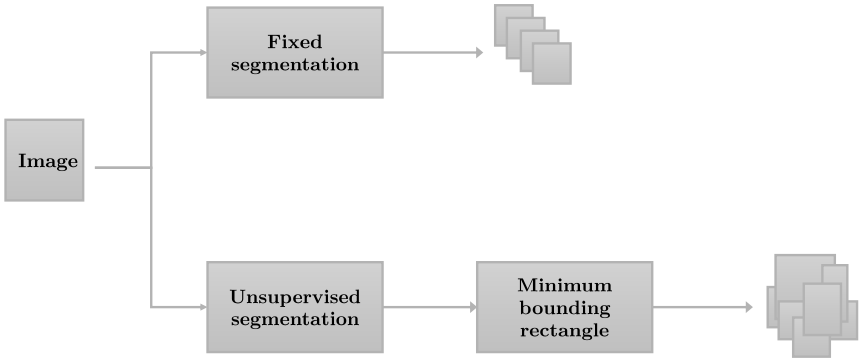


FIGURE 5.27: Taking variable or fixed size patches of an image.

several and—as already said— it is related to pre-processing steps and patch extraction from input images.
Theoretically this is strictly a problem of feature descriptors. Statistical-based features should be less affected from these differences than more structural-based features as LBP. When an image is divided by fixed-size windows there is a general lower warranty that the pattern inside will be homogeneous in comparison of the variable-size case that may have more adherence to the actual image structure.
Fixed-size patches have been tested with three more dataset, so slight variations can be due also to this fact. After reviewed all specific performances about using one type of patch instead another here the objective is to summarize the results obtained and to make a comparison between them.
From the results of all the experiments we can observe that performance of both methods are quite close.
The best results of LBP-based (l) features have found confirmation in both the two cases. For fixed-size patches the l feature set improves their accuracy more than 4%, while for the other two statistical features (f and s) the results in both cases are almost the same (as shown in chart in Figure 5.28).

Analogue tendency is found comparing RGB and gray colour channels. As can be seen in chart in Figure 5.29, the lines shapes are substantially identical.

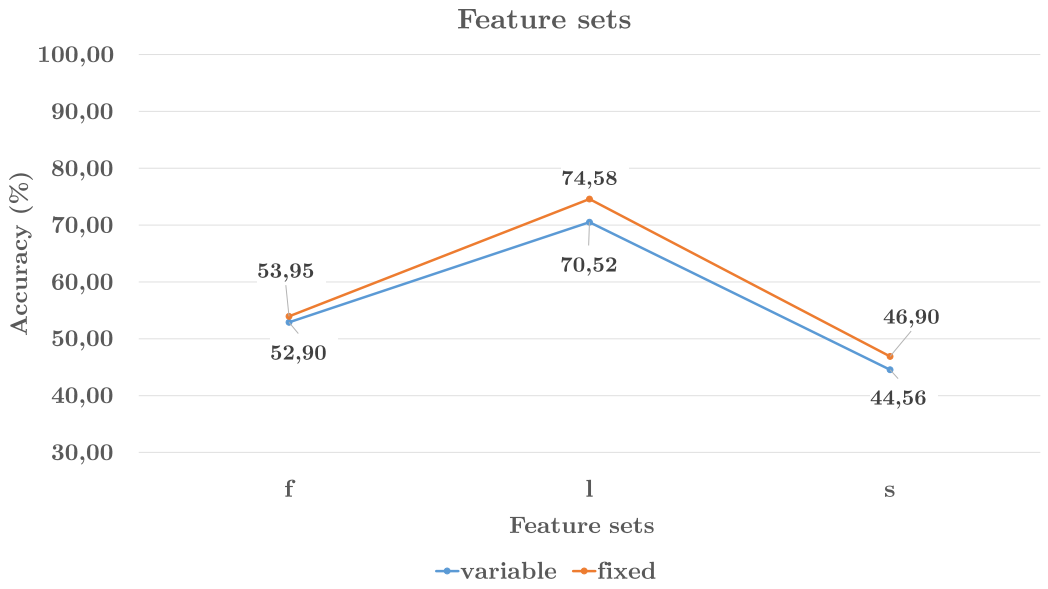


FIGURE 5.28: Comparison between feature performance in relation to the cases of variable and fixed size.

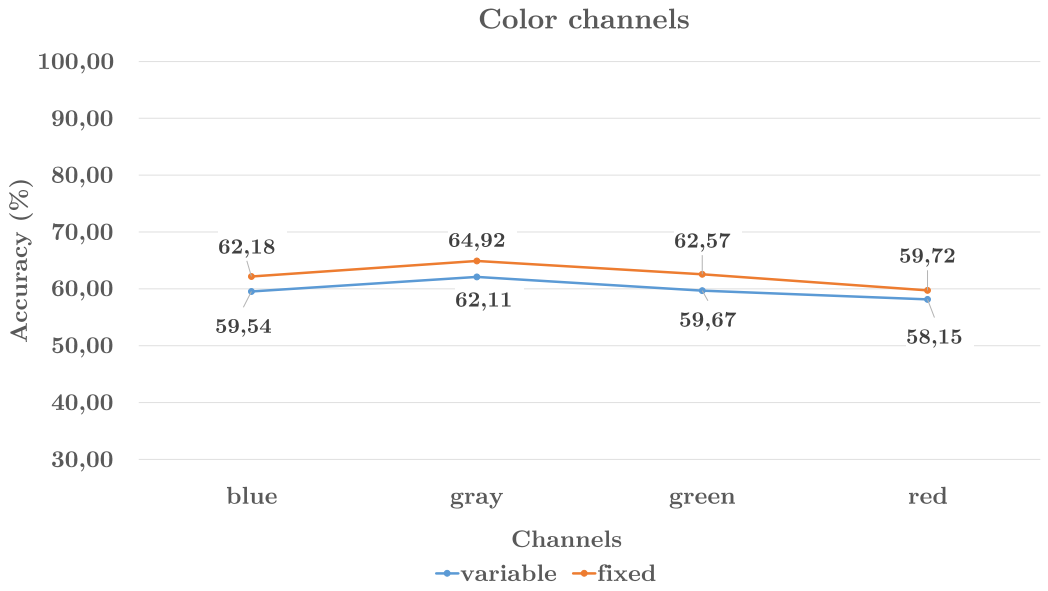


FIGURE 5.29: Comparison between colour channels performance in relation to the cases of variable and fixed size.

Continuing with this analysis we can interestingly compare variable-size versus fixed-size patches for what concerns the nine datasets (D1-D9) in common.

The chart in Figure 5.30 shows appreciable differences than the previous ones. All the lines have roughly the same shape, but on datasets D2, D3 and D4, variable size patch achieves better results, while fixed size configuration do it on dataset D6.

FIGURE 5.30: Comparison between datasets performance in relation to the cases of variable and fixed size.

Slight differences in performance might be justified from the fact that the image patches extracted from the same dataset might vary in consistency in relation to the class to they belong. To limit this issue, patches have been extracted carefully from the same image areas both in case of fixed size and variable size window. Furthermore we remark that previous results are reported as aggregated set, averaged over a wide number of experiments and examples, so this statistically should mitigate non-biased variations in patches extraction methods.

For these reasons we argue that differences achieved in dataset have to be primarily related to the environment that a dataset represent.

In dataset D2, for example, we noticed that (relative) small amount of errors was caused by classes that often appear together in the same patch. The difficult is inherently linked to the environment and in these cases the variable-size patches can better fit images. An example is in Figure 5.31 where the spotted vegetation is mixed to sand. Fixed-

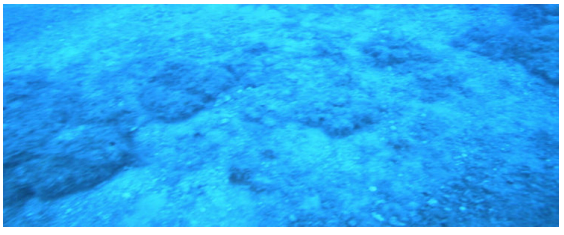


FIGURE 5.31: An example of spotted vegetation mixed to sand with additional colour limitations.

size sampling works better when the classes in the scenario appear well separated and

uniform.

In relation to the configuration parameters, the employed feature set and input colour channel, the SVM kernel distinction is not as much relevant. Again, there is a slightly overall better performance (about 3%) in using fixed-size patches in combination with all kernels.

To summarize, there is no a clear evidence to definitely conclude if it is clearly better using fixed- or variable-size image patches from feature extraction. Descriptor differences do not seems in both cases significantly influence the performance. Nevertheless what we may say is that in the case of absolute ignorance about the environment, the fixed patch may be preferred because their simplicity and because they allow anyhow to gain, on average, some percentage point of classification accuracy.

Otherwise if we know that the environment under analysis is characterized by a low class-uniformity, the variable-size patch extraction might be the right choice.

5.3 Best performance over dataset

Until now have been discussed results by averaging them with respect to main considered parameters, as feature sets, colour channel and SVM kernels.
Table 5.29 and Table 5.30 reports the best performance achieved on every dataset, respectively for fixed-size and variable-size patches. Tables reports mean values achieved

TABLE 5.29: Best results achieved in every single dataset D01-D12 (with relative configuration) [f]

Data	Ch	Feat	Ker	Classes	Accuracy	(top)	(bottom)
D01	gray	l	linear	5	87,09	88,61	83,54
D02	green	l	RBF	5	76,78	79,66	69,49
D03	green	f	RBF	3	79,17	80,56	76,39
D04	gray	f	poly	2	99,03	100	98,39
D05	green	f	RBF	3	98,47	99,1	97,97
D06	blue	l	poly	4	98,89	98,89	98,89
D07	blue	l	poly	4	96,33	96,75	95,86
D08	gray	l	poly	4	81,63	88,78	72,45
D09	blue	l	RBF	3	94,07	96,3	90,74
D10	green	l	poly	2	99,36	99,54	99,24
D11	green	f	RBF	3	94,31	94,9	93,42
D12	blue	l	poly	3	93,25	95,69	88,24

TABLE 5.30: Best results achieved in every single dataset D01-D09 (with relative configuration) [v]

Data	Ch	Feat	Ker	Classes	Accuracy	(top)	(bottom)
D01	gray	l	linear	5	66	70	60
D02	gray	l	poly	5	96,18	98,18	92,73
D03	blue	l	poly	3	87,16	91,04	83,58
D04	gray	f	poly	2	98,44	100	97,4
D05	blue	f	RBF	3	98,19	98,61	97,68
D06	green	l	poly	4	90,98	95,12	87,8
D07	gray	f	RBF	4	93,64	95,33	91,59
D08	gray	l	linear	4	81,32	84,62	78,02
D09	red	f	RBF	3	94	100	86

with *k-fold* validation, with also the better and worst results obtained in each execution. Both in fixed and variable cases we can observe that there is not dependence on colour channels, unless a little preference for gray-level in the variable case.
For what concerns kernels there is a slight preponderance of polynomial compared to the RBF kernel. Despite its global results in each table appears almost a situation in which the best configuration is the linear kernel; for example it is the case of variable-size D1 and D8 datasets. It is interesting that in all these cases the performance are under the average (less than 90 %) on all datasets. We may argue that when other kernel are not

able to well discriminate, the linear one become the best choice.
The most powerful feature set seen—not surprisingly—is the uniform LBP that in the case of fixed-size patches appear in approximatively 70% of the best dataset performance. In the chart in Figure 5.32 are summarized all the best configurations.

For dataset D1 to D9 the comparison shows that although accuracy values are in this

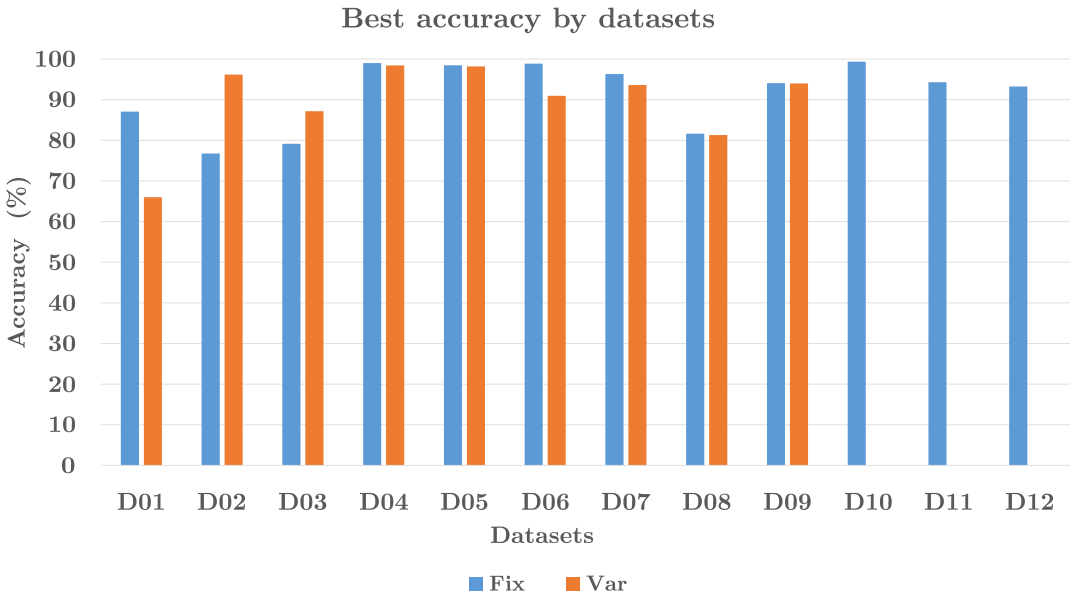


FIGURE 5.32: Best performance obtained in each dataset.

case higher than those obtained on averaged measures over datasets (see section 5.2.10), they are substantially the same.
Referencing again the two Tables 5.29 and 5.30, we can observe that higher results (close to 100%) have been achieved with the 2-class datasets D4 (variable and fixed cases) and D10 (only fixed case). Although this fact was expected, similar high and comparable results are achieved also with dataset with more classes as D5 and D6.
To conclude, Table 5.31 shows the difference in best accuracy for fixed and variable windows. In almost all scenarios a fixed-size patch seems to be preferred, even if the difference might be sometimes limited. Datasets D1 and D2 are exceptions where the choice of the method is a more performance-critical selection.

5.4 Qualitative results and discussion

The purpose of this section is to provide a qualitative and concise summary of obtained results regarding our test for underwater classification. In particular our aim is to

Table 5.31: Accuracy spreads for each dataset

Dataset	**Accuracy**	(Fix/Var)
D01	-21,09	Fix
D02	19,40	Var
D03	7,99	Var
D04	-0,59	Fix
D05	-0,28	Fix
D06	-7,91	Fix
D07	-2,69	Fix
D08	-0,31	Fix
D09	-0,07	Fix
D10	-	-
D11	-	-
D12	-	-

give a short answer to the main questions, all related to the task of underwater scene classification:

1. Which is the descriptor—from those employed—that provides the best performance in underwater scenarios?

   The LBP-based descriptor is the one that shows best performances over all experiments. All the three descriptors have a similar nature, but we see that using a more structural based approach gives us better results. It has to be considered, also, that describing image patches by uniform LBP has itself a statistical nature (see 4.2.3), precisely when we consider for every patch its histogram distribution. Although we made experiments on a manifold image sets—with the objective to take many different environment—they cannot be considered comprehensive of all possible scenarios.
   Looking at best results on single experiments, other descriptors than LBP-based may locally achieve better results, even if they are never widely greater. From our tests there are not cases that totally discourage the LBP features and we also noticed that they perform sharply better than others when more classes in a dataset have to be distinguished. First and second order statistical-based features are inclined to have a higher generalization and are not able to correctly discriminate between classes.
   For all these reasons the LBP-based descriptor seems to better get the intrinsic characteristics of the underwater environmental appearance in relation to other tested feature sets.

2. What are effects to feed the classification algorithm with fixed-size or variable-size image patches?

   As we saw, different image segmentation methods, with variable or fixed size, may affect the overall performance. In particular while the global performance are less or more the same, on a single dataset this value may vary considerably. From our tests, the approach with fixed window patch should be preferred for its effectiveness and simplicity.

3. Are there significant variations in using different colour channels?

   The answer to this question is no, at least considering the best results for each dataset. From the overall performance analysis we noticed that—as might be supposed considering the underwater scenario and its light transmission properties—the red channel is the one that give, on average, the lowest accuracy. In particular in only one experiment the red channel appeared as the best choice and not surprisingly in a dataset with a good illumination. Other experiments are distributed almost uniformly over all the remaining colour channels, and just a slight preponderance for the gray-level is present.

4. Which is the best SVM configuration for classification?

   Polynomial and RBF kernels are doubtless the best choice in comparison of the majority of experiments. Performance of both are comparable and there is not a clear prevailing one.
   In some datasets, instead, the linear kernel—when used in combination with LBP features—may achieve the best accuracy results. Is interesting to observe that on dataset where linear performs better, the accuracy is significantly lower than those of other image sets. This might be due to the dataset itself that may have classes that are very close and in comparison of non-linear approaches—apparently unable to catch these differences—the easiest separation of linear kernel emerges as the best choice.

5. What are datasets that achieve better/worst performance?

   In all datasets, as shown also in chart in Figure 5.32, an accuracy widely higher than 90% may be achieved with a proper configuration. The exceptions are datasets D1, D3 and D8 that, even considering the best configuration, have lower values (anyhow they are all over the 80%).
   Even from further specific analysis, apparently these three datasets haven't any visible shared characteristics so we cannot hypothesize a common cause behind. Also the number of classes seems do not explain these performance differences

although they show a globally (expected) descending trend of accuracy as this number increase.

6. Which are classes that show better performance?

   Which are the better classified classes and those are instead highly misclassified is one interesting question to answer, and somehow related also to the previous ones. Working with a dozen of different environmental datasets makes obviously this process quite complicate for a punctual case-by-case treatment.
   Table 5.32 reports the resulting confusion matrix with the performance achieved over all datasets. This table has been built considering the best performance

Table 5.32: Confusion matrix in relation to all classes (% values)

	algae	**coral**	**h-veg**	**sand**	**vegn**	**archaeo**	**water**	**rock**	**unkwn**
algae	6,10	0	0	0,07	0,31	0,07	0		-
coral	0,03	15,12	0,03	0,03	0,07	-	-		-
h-veg	0	0,24	0,65	0	0,14	-	-		-
sand	0,03	0,07	0	25,72	0,10	0,58	0,10	0,17	-
veg	0,17	0	0	0,17	9,81	0,07	0	0,51	-
archaeo	0	-	-	0,10	0,31	12,60	0,03	-	0
water	0	-	-	0	0	0	10,25	-	0,20
rock	-	-	-	0,37	0	-	-	11,34	-
unkwn	-	-	-	-	-	0	0	-	4,43

over all (the twelve) datasets. Here, although all the represented classes are not balanced due to non-uniformity across datasets, it can give an interesting quick look to the global behaviour. In particular we see from the diagonal that more than 95% of examples are in total well classified. Classes that appear more frequently misclassified are:

- *archaeological* vs *sand*
- *algae* vs *vegetation*
- *rock* vs *vegetation*
- *high vegetation* vs *coral*

Considering the same class in different datasets might be considered misleading. In fact, by changing dataset, the concept and the appearance, represented by a class name can substantially vary.
We conducted experiments also by combining available data carried out from different datasets. For these image sets taken from the same environment or from one with a close affinity, performance resulted comparable to those obtained with more consistent datasets. When differences become greater the appearance of patches, even if they are labelled with the same name, might be characterized by strong

dissimilarities, making extremely difficult the classification.

For example, the *archaeological* class, corresponding to objects lying on the seabed, may have a shape that is easy to be confused (see Figure 5.33), sometimes also by human intelligence, without a more contextual knowledge.

Classes that we considered for classification have not a well defined shape or

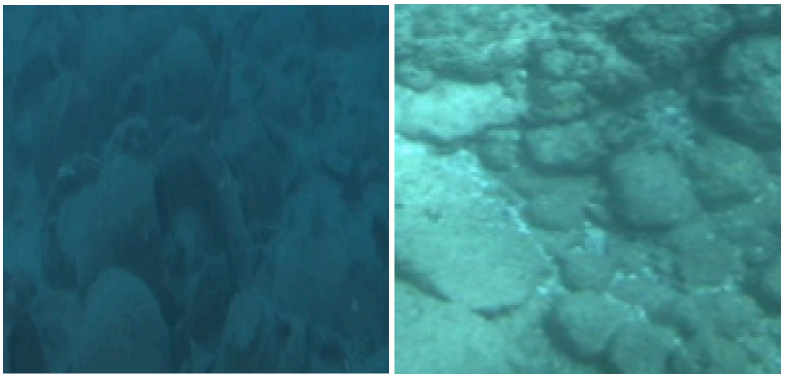


FIGURE 5.33: Example of similar appearance between classes. Left image is from class *archaeological* while the right one is from *rock* class.

peculiarities. Terrestrial natural scenes share similar problems, but in the case of underwater scenarios, environmental conditions are worst because, other than shapes, we also cannot trust on colours.

Chapter 6

A new feature descriptor for underwater image classification

As shown in Chapter 5 the Local Binary Pattern feature set is the one that achieves best performance with all the underwater dataset considered. Anyhow LBP is a general purpose approach with large employment in problem like texture analysis, motion detection and face recognition ([145]).
The original idea behind LBP already caused several improvements and extensions, depending on the particular application field. In our classification tasks we adopted the *uniform LBP*.
Starting from this, the focus of the first part of this chapter is to show the development of a new version of LBP, specifically aimed to be used in underwater environment. For this reason we called it *underwater LBP*.
In the second part of this chapter the results obtained over all our datasets are presented and compared with the classic LBP, underlining its strengths and (potential) weaknesses.

6.1 Motivations

The underwater environment present some peculiarities that might hardly interfere and must to be handled in comparison of the classic terrestrial acquired images.
In Chapter 1 we dealt with the characteristics proper of the water medium that directly affect the process of acquiring images. Regarding the problem of segmenting/classifying an image, the scattering effect—mostly due to the presence of dispersed particles or colloids— is the one that might determine the highest effects.
Despite the dehazing algorithm presented in Chapter 3, the scattering effect cannot be

in many case completely removed. For this reason we define a descriptor that is more robust and reliable in relation to this phenomenon.
Note that from tests on some of our dataset using the uniform LBP features over a pre-dehazed image does not change substantially the classification performance and an haze-recovered image—although its improved appearance—does not seem to increase the robustness in the classification task. Dehazing algorithms are mostly focused on image appearance and do not provide reliability when lighting conditions vary in the same scene. In addition to that the non-linear transformations imposed also by the transmission refinement may decrease the consistency of this approach.
In synthesis, what we want to perform is a direct improvement of the tolerance to underwater distortions, intrinsically on the binary pattern codification. The focus is again mostly on the scattering effect.
A known drawback of the LBP—as well as in general of local descriptors that make a vector quantization—is that they show a poor consistency: small change in the input image might not have only a small effect in the output. Some solutions propose to replace the thresholding function, inside LBPs, with a smoother one, as for example the approaches reported in [171] and [148].
In their simpler form, the LBPs are substantially constituted by a vector calculated over a neighbourhood of a given reference element (i.e. pixel in straightforward implementation). Here, we avoid to report the extensive theory on binary pattern that can be found on Chapter 4, but we'll be concentrated only on changed aspects.
The LBP vector is computed by making comparisons between the central pixel of a squared (or circular) window one-by-one with all its neighbouring in a counter-clockwise fashion. The LBP is a binary vector and its components are the 0-1 results of this comparison (0-1 if the central pixel is respectively less or higher the considered neighbour). In uniform LBP all the computed binary vectors are then grouped into a predefined set of configurations, called uniform pattern.
Water scattering affect an image covering it with brighter spots of various dimensions, less or more concentrated into the scene.
The effect of scattering on a single LBP is synthetically described in Figure 6.1. For simplicity we consider the general and largely adopted case of $LBP_{8,1}$ which consists in a 3×3 windows with the central pixel surrounded by 8 neighbours. Figure 6.1 reports a correspondent LBP calculated over the same reference central pixel $C = (c_x, c_y)$. In the upper part of figure $((a))$, is represented the case in which the LBP is computed considering the actual radiance of an image, while in the bottom of the same figure is reported how a light scattering event might affect this computation. Some neighbouring pixels may present an increased intensity value due to a close brighter spot as the resulting of a light scatter event. This phenomenon may—as actually does in the figure—change

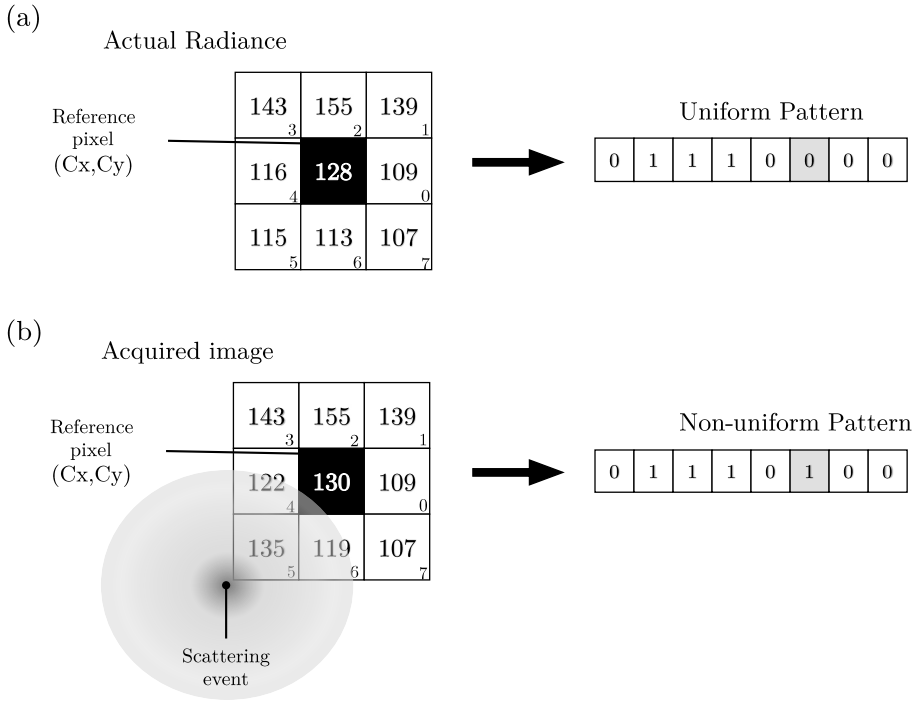


FIGURE 6.1: An example of how scattering events might affect the LBP computation. Figure (a) shows the LBP calculated over a reference point in where there are no distortions in the actual image radiance. In figure (b) instead is reported the same point but now it is close to a brighter spot that partially interferes with the LBP evaluation neighbourhood. Even if the Uniform LBPs are robust in relation to monotonic intensity changes, the scattering effect may cause limited and local intensity variations. As can be seen in this case the resulting binary vector might be different in the two cases. In computing the LBP over an entire image patch, few isolated changes like the previous one are well tolerated, but when they increase to much, as in presence of diffuse scattering, the performance of a classifier based on these features might be seriously affected.

the resulting LBP vector. As can be seen in this case, the result of the comparison with the $5th$ pixel has altered the correspondent vector component which passed from zero to one. This single change, considering the theory behind the *uniform LBP patterns*, makes that the binary pattern associated to the central pixel C, passes from being *uniform* to be *non-uniform*.

An high presence of scattering events leads to have an increasing number of non-uniform patterns, associated to many pixels and consequently causing a general reduction in the discrimination capabilities of this feature set.

In classifying an image, each extracted patch is described by taking the histogram representing the distribution of LBP vector clustered accordingly to a number of pre-defined patterns (in general are 58 uniform patterns plus one non-uniform configurations, as reported in Figure 4.9 on Chapter 4).

Until the amount of scattering is limited, only a little number of LBPs will be distorted

and the effects on final classification might be negligible. In the other hand, when the scattering phenomenon increase, these diffuse changes might hide the real image structure described by LBP. Clearly, working with terrestrial images this problem is less evident because image are in general characterized by a better clearness.

6.2 The underwater LBP

As the *Dark Channel Prior* theory explains (see section 3.2.1), in hazy/scattering situations the more reliably image values, compared to the actual radiance, are those closer to zero.

For short, by keeping the underlining architecture of *uniform Local Binary Pattern* we changed the way in which the vector is computed to better work with underwater images, or more general, with images affected by similar effects.

Like classic LBP, our approach computes a binary vector for all pixels on a given image patch. It starts by considering a central reference pixel $C = (c_x, c_y)$ and its circular neighbourhood.

In the following, the minimal case is described, but the neighbouring set may be in-

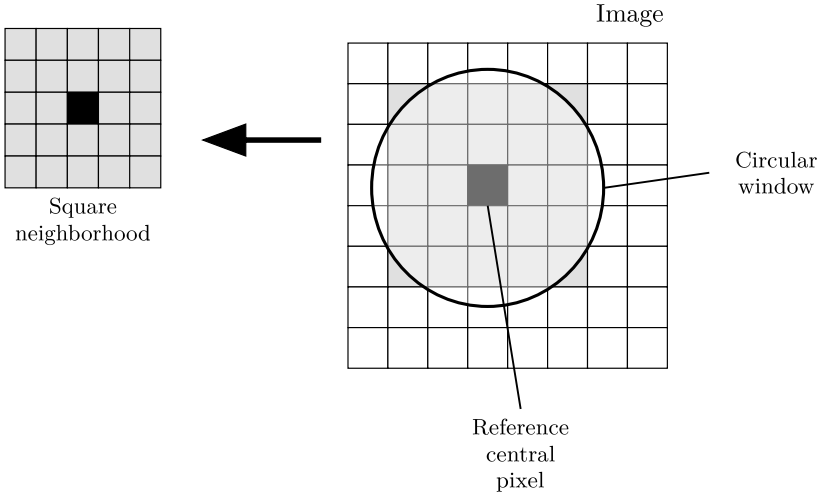


FIGURE 6.2: The neighbourhood area where the Underwater LBP is computed is theoretically defined as a circular surrounding of a given radius (in px). In practice for actual computation, instead to interpolate the intensity values, the entire enclosed region is taken.

creased depending on particular employments. Sometimes, for the general use of LBP

might be a better solution to properly scale the input image instead to increase the neighbourhood size of the reference pixel. The concept of *circular neighbourhood* is useful mostly to theoretically explain this approach and its extensions than in practical uses where, considering pixels as fundamental units, a squared windows is considered instead to actually interpolate them (Figure 6.2).
The minimal window size for computing our underwater LBP is 5×5 pixels.
All the information inside this window is codified in a 8-bit binary vector, associate to the central reference pixel. Each ordered component of this vector, starting from left, is numbered incrementally from $p0$ to $p7$ (Figure 6.3) and is computed as following. Considering a reference image coordinate system as reported in Figure 6.4 and with the

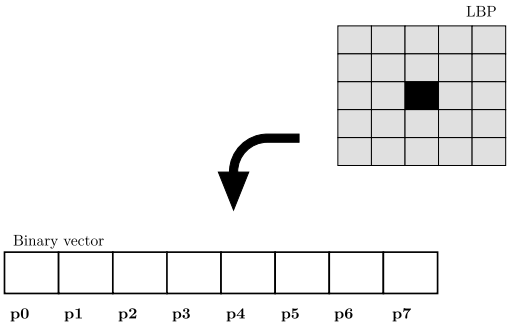


FIGURE 6.3: The information contained in a 5×5 uwLBP window give rise to an ordered binary vector with components labelled as $(p0, p1, \ldots, p7)$.

origin located at the top-left corner, we firstly isolate the neighboring pixels that lie on the X and Y axis. There is a total of 4 couples of adjacent elements with at least one side in common, starting from east and proceeding counter-clockwise and respectively indicated as e_0, e_2, e_4, e_6. Inside every pair the minimum value is taken so with respect to the reference pixel C in the Figure 6.4 we have:

$$\begin{cases} e_0 = \min\left(p(c_x, c_y + 1), p(c_x, c_y + 2)\right) \\ e_2 = \min\left(p(c_x - 1, c_y), p(c_x - 2, c_y)\right) \\ e_4 = \min\left(p(c_x, c_y - 1), p(c_x, c_y - 2)\right) \\ e_6 = \min\left(p(c_x + 1, c_y), p(c_x + 2, c_y)\right) \end{cases} \tag{6.1}$$

where $p(\cdot)$ are the image intensity values, over a single channel.
In relation to neighbourhood in diagonal position (Figure 6.5) all the 4-elements that compose the squared region are considered.
In the same way as before, for each group indicated respectively as e_1, e_3, e_5, e_7, starting

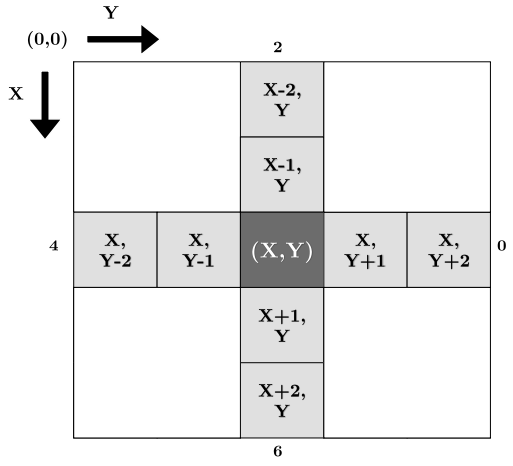


FIGURE 6.4: In computing the binary vector of b 5×5 uwLBP, ar firstly considered the neighbouring elements (around the reference central element) that lie on the X and Y directions. In particular 4 couples of near elements are identified and each one is associated to one component of the final binary vector. The numbers $0, 2, 4, 6$ indicate the position of the correspondent component in $(p0, p1, \ldots, p7)$ vector.

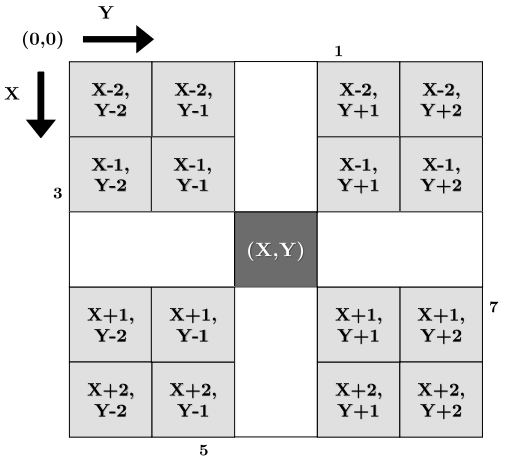


FIGURE 6.5: In the 5×5 uwLBP diagonal neighbours are taken 4-by-4 and each group is related to the component $p1, p3, p5, p7$ of the resulting binary vector, respectively assigned starting from eastern group and proceeding counter-clockwise. Inside each single element are reported its coordinates in relation to the central reference point $((x, y))$.

from the north-east element the minimum value is taken. We formally obtain:

$$\begin{cases} e_1 = \min\left(p(c_x - 1, c_y + 1), p(c_x - 1, c_y + 2), p(c_x - 2, c_y + 1), p(c_x - 2, c_y + 2)\right) \\ e_3 = \min\left(p(c_x - 1, c_y - 1), p(c_x - 1, c_y - 2), p(c_x - 2, c_y - 1), p(c_x - 2, c_y - 2)\right) \\ e_5 = \min\left(p(c_x + 1, c_y - 1), p(c_x + 1, c_y - 2), p(c_x + 2, c_y - 1), p(c_x + 2, c_y - 2)\right) \\ e_7 = \min\left(p(c_x + 1, c_y + 1), p(c_x + 1, c_y + 2), p(c_x + 2, c_y + 1), p(c_x + 2, c_y + 2)\right) \end{cases}. \tag{6.2}$$

There are in total 8 ordered values $(e_0, \ldots, e_7)$ that represent the neighbourhood of the central pixel C, as reported in Figure 6.6. It is possible to see that this configuration is

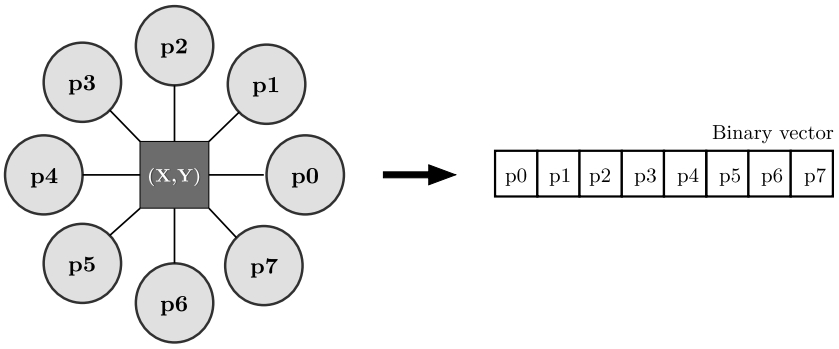


FIGURE 6.6: The complete relation between values computed on every neighbouring group of the central element and the actual position in the final binary vector.

in analogy with the classic LBP. Each e_i element is compared with the central pixel C and the result is assigned to the corresponding p_i position. In particular:

$$p_i = \begin{cases} 1 & if \quad (e_i - e_C) \geq 0 \\ 0 & if \quad (e_i - e_C) < 0 \end{cases} \tag{6.3}$$

where the e_C is the intensity value of the reference central element $C = (c_x, c_y)$. The resulting binary vector is finally $(p_0, p_1, \ldots, p_7)$.

Following the theory behind the *uniform LBP* ([9] and [10]) this last vector is then assigned to one of the 59 possible patterns as explained in Figure 6.7. In particular, if the number of 01/10 transitions inside the binary vector, is equal or less than two, the vector is associated to one of the 58 possible *uniform* configuration depending on its actual values. Otherwise if the number of 01/10 transitions is more than two, the binary vector will be assigned to the single *non-uniform* class.

Each LBP describe a point and the cumulated histogram distribution obtained from binary vectors computed over all the pixel of a given image patch represents the final descriptor for it.

Summarizing, Underwater LBPs share the same approach of classic uniform binary pattern (in particular the one defined on Chapter 4) but are differently computed. They have substantially the same invariance properties, but other than the classic uniform

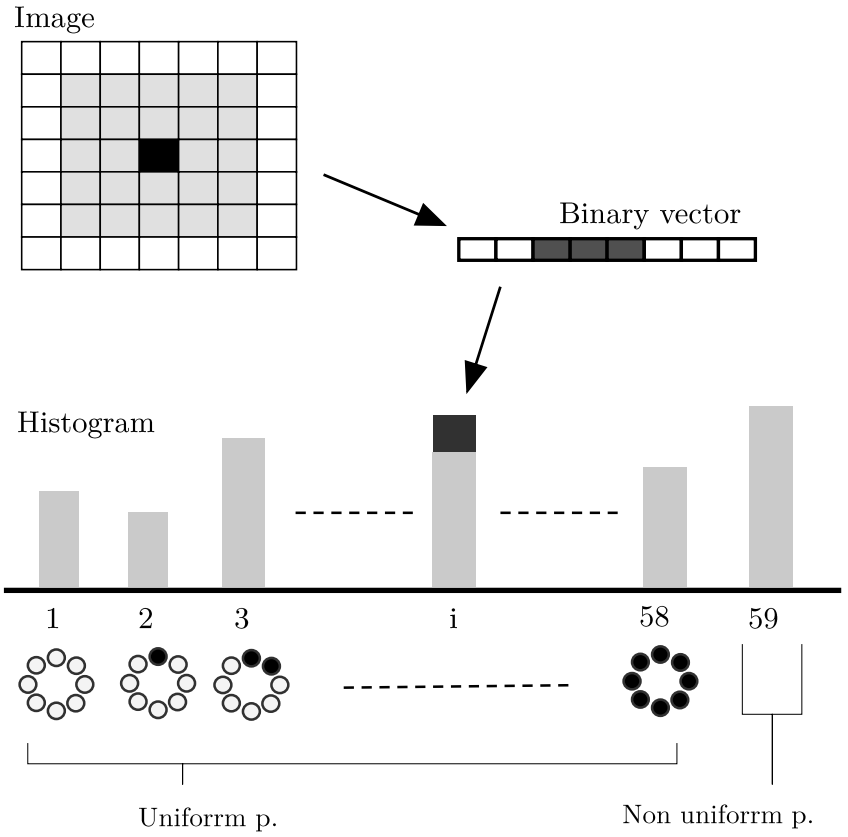


FIGURE 6.7: Each binary vector extracted at every position of an input image, contributes to the creation of an histogram actually representing the complete distribution of patterns inside the patch. Binary vectors are clustered in 59 total patterns; 58 of them are the so called *uniform* patterns (see $LBP^{u}_{8,1}2$ in [9]) and correspondent to a binary vector with two or less 01/10-transitions inside. All other possible binary configurations are all grouped in a single bin, the 59th, which represents all the *non-uniform* pattern. The resulting histogram is finally the actual feature set that describe the given input image patch.

LBP, this underwater version is more reliable in finding uniform patterns in presence of spotted light inconsistencies. By using the minimum values of the reference pixel neighbourhood, each time is selected the one that theoretically would be less distorted in comparison to the actual radiance.

The uniform configurations are those that have more importance and our proposed method allow to better catch them. As it will be shown in next section, our improvements make the final descriptor more robust for classification in underwater images. Wherever degradation effects induced by scattering have comparable dimensions with the binary pattern elements the best results are observed.

Until this approach has been described using the smallest 5×5 configuration. However depending on image and its expected degradations, an extended version can be considered; it has anyhow must be taken into account that by increasing the neighbourhood

window also the computational cost consequently will increase. Without any particular implementation the theoretical cost of computing the underwater LBP descriptor is $O(N * m)$ where N is the number of pixel in the input image and m is the number of pixel in the window used to calculate the binary vector. Furthermore, instead using single pixels as the basic elements also bigger segmented areas might be employed. As well as the classic LBP, this descriptor might be easily scaled.

6.3 Test and results

We initially have conducted several test to actually prove the underlying idea behind this new feature set (i.e. a more robust behaviour due to the reduction of the number of non-uniform pattern carried out by this new proposed feature set).
We analysed the feature vectors extracted both with uniform LBP and our underwater LBP evaluated on the same image patches.
We found empirical evidence that the amount of non-uniform patterns in the uwLBP feature vectors is in general lower, especially on those images with highest scattering phenomena.
For example, taking all the images present in dataset D12 (Figure 6.8)—one of the haziest dataset— the 77.8% of examples presents an effective reduction of the component corresponding to non-uniform patterns. Compared to the uniform LBPs the reduction in each example is on average near the 24%.

In the following of this section we compare the results obtained from testing the *Underwater LBP* (uwLBP) with respect to our datasets previously defined in Section 4.3 of Chapter 4. We use as comparison data obtained with *Uniform LBPs* (simply LBP in the following and specifically considering the $LBP^{u2}_{8,1}$) for almost two evident reasons. Secondly the underwater LBP shares some properties with the classic binary pattern and it is interesting to see if and how this new version leads to better performance.
The test conducted below follows the approach used in Chapter 5 and, if not otherwise specified, also the notation will be kept consistent. In particular, questions regarding the classification scheme adopted, the configuration and parameters used may be found in that chapter. In the same way, datasets description, composition and peculiarities may be found in Chapter 4 and will not be here reported again.
By keeping the same datasets and the same framework above, here the experiments are focused to analyse a limited subset of parameters. Now, we will use only those configurations that gave the best overall results.

FIGURE 6.8: Example of an image taken from dataset D12.

Overall, *graylevel* images was those that led to better performance and here are chosen—if not otherwise specified—as the main channel input. The *Polynomial* and *Radial Basis Function*, that achieved best results in previous experiments, are now the only considered and maintaining the same configuration as before (see section 4.4.2).
The distinctions about the two cases of variable- versus fixed-size input image patches is also kept in the current evaluation.
To summarize, experiments are now carried out with: 1) *12 datasets* (reduced to 9 in the case of variable-size inputs); 2) *polynomial* ("poly") and *radial basis function* ("RBF") kernels; 3) *LBP* ("l") and *uwLBP* ("u") features. Every single classification test has to be intended as the mean of multiple cross-validated executions. The *accuracy* is again the principal adopted evaluation and measure employed for comparisons.

6.3.1 Features: Overall

Table 6.1 and Table 6.2 report the obtained results by using LBP or uwLBP, respectively for variable and fixed size patches. The results are averaged on all the datasets and SVM-kernels used.
It is possible to see that in both situations the new uwLBP perform better.

TABLE 6.1: Features accuracy (mean and standard deviation) - [v]

Features	**Accuracy**	**(std dev)**
l	85,64	9,85
u	86,46	8,94

TABLE 6.2: Features accuracy (mean and standard deviation) - [f]

Features	**Accuracy**	**(std dev)**
l	86,62	8,30
u	87,73	6,65

Using variable window size the difference is limited to less than one percent, while is higher in the case of inputs consisting in fixed size patches.
Giving a look to the related standard deviations it can be noticed that our proposed descriptor presents lower values in all cases. This means that considering the best (or worst) performance, by employing one feature set instead of another may lead to the same extremal values even if the uwLBP is a priori preferable.
In term of absolute results, using fixed size patches led us to achieve in general better results than the variable size ones.

6.3.2 Features and SVM kernels

The relation between the uwLBP features in relation to the polynomial or RBF kernels are reported on Table 6.3 for the variable size and in Table 6.4 for the case of fixed size. What emerge is that using a kernel instead of another does not change significantly the

TABLE 6.3: Overall accuracy of features by kernels (mean and standard deviation) - [v]

Features [by kernel]	**Accuracy**	**(std dev)**
l	**85,64**	
poly	85,71	10,69
RBF	85,56	9,59
u	**86,46**	
poly	86,91	9,30
RBF	86,01	9,10

performance. This, was an expected result considering what has been said in Chapter 5. Even if the polynomial kernel appears to be better in three out of four cases it must to be noted that the uwLBP feature used with the RBF obtains the best overall performance (with fixed windows size).
Furthermore is remarkable that the uwLBP outperforms the LBP also considering each kernel individually.

TABLE 6.4: Overall accuracy of features by kernels (mean and standard deviation) - [f]

Features [by kernel]	Accuracy	(std dev)
l	**86,62**	
poly	87,06	8,07
RBF	86,18	8,86
u	**87,73**	
poly	87,43	6,61
RBF	88,03	6,97

6.3.3 Features and Datasets

Table 6.5 shows results obtained on all of the 9 classified dataset with inputs of variable window size. Looking at the relative chart (Figure 6.9) the shape of both lines show a

TABLE 6.5: Overall accuracy of feature sets by datasets (mean and standard deviation) - [v]

Feature sets [by datasets]	Accuracy	(std dev)
l	**85,64**	
D1	63,30	1,84
D2	96,18	0,00
D3	83,58	2,53
D4	95,58	0,00
D5	81,84	0,76
D6	90,25	0,69
D7	88,69	1,19
D8	80,33	0,16
D9	91,00	2,55
u	**86,46**	
D1	67,60	1,70
D2	93,91	0,13
D3	89,55	0,42
D4	95,84	0,74
D5	77,22	4,33
D6	91,95	0,00
D7	90,38	1,72
D8	82,31	0,47
D9	89,40	0,85

substantial consistent behaviour.

The higher spreads can be found on dataset D1, D3 and D5, where respectively the uwLBP for the first two and the LBP for the last one perform better. In all other cases there is a general preference for the uwLBP even if the differences are under 2%, and so

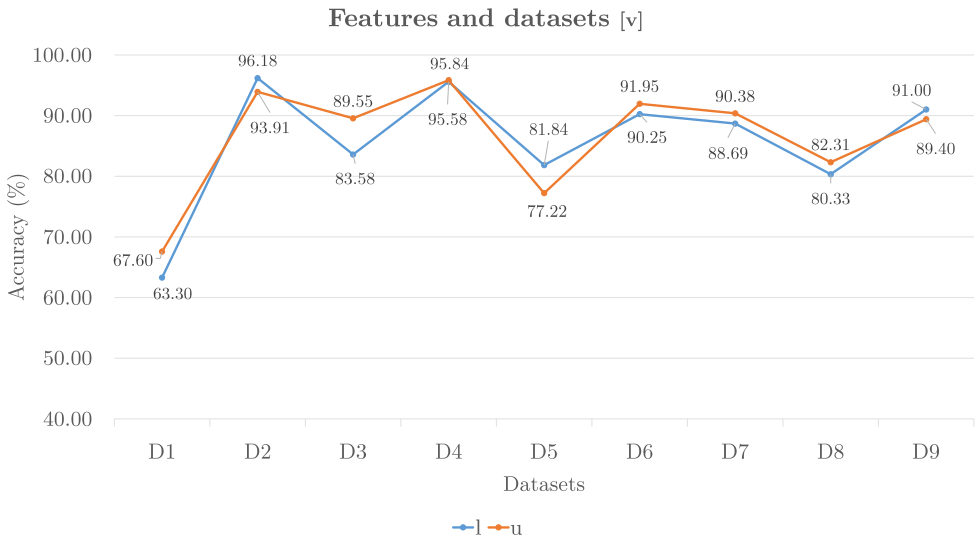


FIGURE 6.9: Overall accuracy of features with respect to the 9 datasets and input patches of variable size.

they appear quite close.

Grouping datasets by the number of represented classes (see chart on Figure 6.10) it can be observed that the uwLBP seems to work better than classic LBP with higher number of classes. Switching to the case of fixed size window input Table 6.6 and the

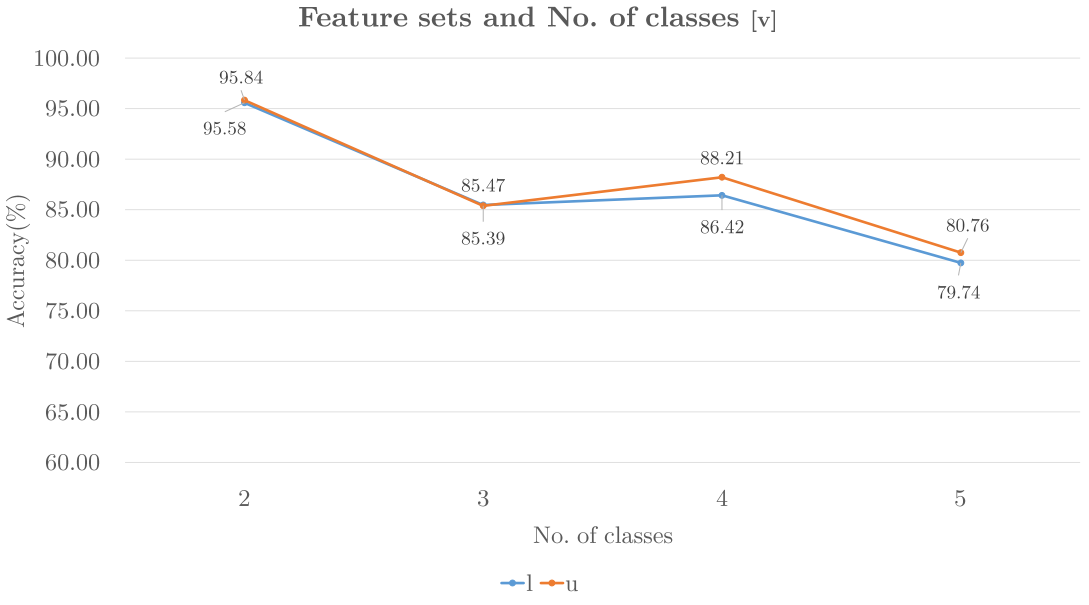


FIGURE 6.10: The accuracy performance variations with respect to the number of classes inside the datasets. (Variable window size)

related chart 6.11 reports the observed performance on all 12 datasets that we created.

TABLE 6.6: Overall accuracy of feature sets by datasets (mean and standard deviation) - [f]

Features [by datasets]	**Accuracy**	**(std dev)**
l	**86,62**	
D01	85,63	1,34
D02	73,56	0,24
D03	72,78	0,78
D04	88,07	0,46
D05	81,37	4,03
D06	97,89	0,16
D07	92,19	0,00
D08	80,31	1,87
D09	92,22	1,05
D10	98,97	0,08
D11	88,74	0,21
D12	87,73	0,72
u	**87,73**	
D01	87,28	0,45
D02	78,48	2,16
D03	74,72	1,97
D04	88,71	0,00
D05	84,44	1,20
D06	97,78	0,63
D07	89,17	1,26
D08	83,68	3,17
D09	90,00	1,05
D10	96,72	2,45
D11	91,43	1,37
D12	90,32	0,39

Line shapes are close but it can be noticed that those associated to the uwLBP are in general higher. Precisely, there are three dataset where the classic LBP works better and they are D7, D9 and D10.

In a way compatible to the largest standard deviation of LBPs, from the chart in Figure 6.11, it can be observed that they are able to achieve both the best and the worst performance over all datasets.

Considering the number of classes contained by every dataset, using input patches of fixed-size doesn't change what it has been seen in the variable window case where uwLBP continues to work well with dataset composed by more than two classes (see chart in Figure 6.12).

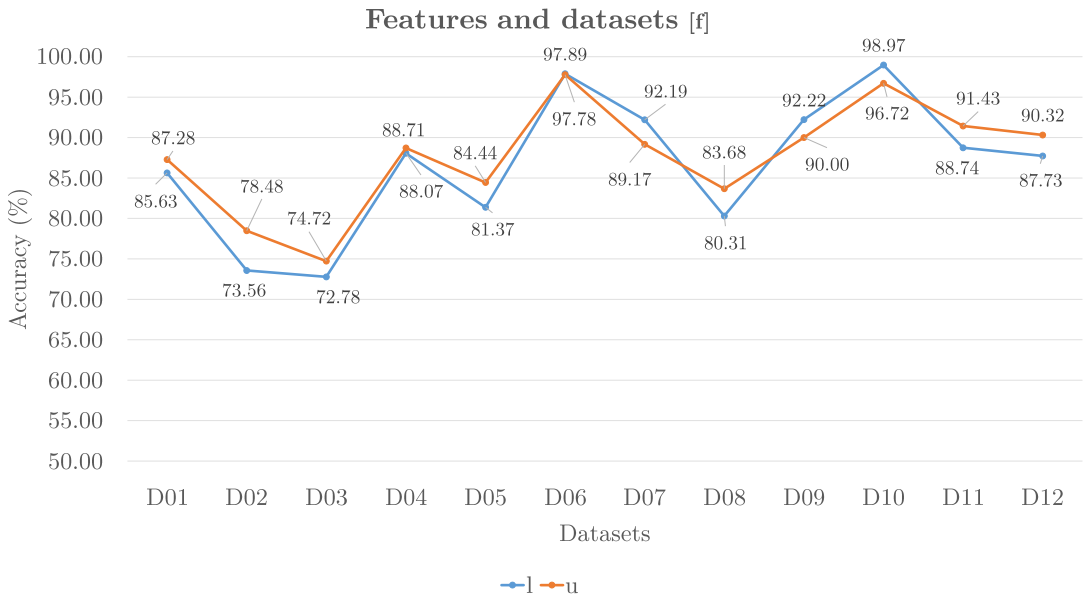


FIGURE 6.11: Overall accuracy of features with respect to the 9 datasets and input patches of fixed size.

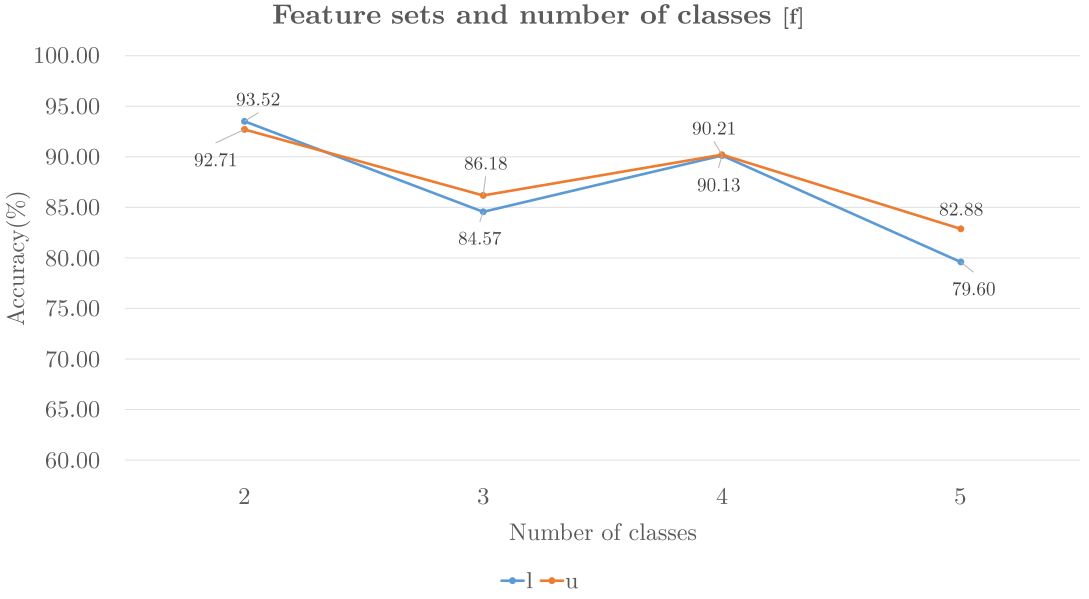


FIGURE 6.12: The accuracy performance variations with respect to the number of classes inside the datasets. (Fixed window size)

6.4 Results: discussion

The previous tables and charts that have been briefly shown are based on averaged results that we obtained. Now we analyse what happens by considering only the best (cross-validated) result. In other words we are showing the expected performance in using those classifiers that achieved higher results on each one of the twelve datasets. Table 6.7 and Table 6.8 reports the best performance achieved respectively with polynomial and RBF kernel for training the SVM.

Table 6.7: Best accuracy with polynomial kernel - [f]

	POLY		
Dataset	**u**	**l**	**(u-l)**
D01	87,59	86,58	1,01
D02	76,95	73,73	3,22
D03	76,11	73,33	2,78
D04	88,71	87,74	0,97
D05	85,29	84,22	1,07
D06	98,22	98,00	0,22
D07	90,06	92,19	-2,13
D08	81,43	81,63	-0,20
D09	89,26	91,48	-2,22
D10	94,98	99,03	-4,05
D11	90,46	88,59	1,87
D12	90,04	88,24	1,80
mean	**87,43**	**87,06**	**0,36**

Table 6.8: Best accuracy with RBF kernel - [f]

	RBF		
Dataset	**u**	**l**	**(u-l)**
D01	86,96	84,68	2,28
D02	80,00	73,39	6,61
D03	73,33	72,22	1,11
D04	88,71	88,39	0,32
D05	83,59	78,52	5,07
D06	97,33	97,78	-0,45
D07	88,28	92,19	-3,91
D08	85,92	78,98	6,94
D09	90,74	92,96	-2,22
D10	98,45	98,91	-0,46
D11	92,40	88,88	3,52
D12	90,59	87,22	3,37
mean	**88,03**	**86,18**	**1,85**

We can see that in both the SVM configuration the uwLBP better performs, and using it in combination with RBF kernel the improvements are greater.
In three out of the twelve datasets (precisely D7, D9 and D10) the classic LBP achieves higher accuracy, while the uwLBP instead is clearly the best choice in dataset D3, D11 and D12.

Taking a look inside these datasets (see Figure (6.13) it is possible to note that actu-

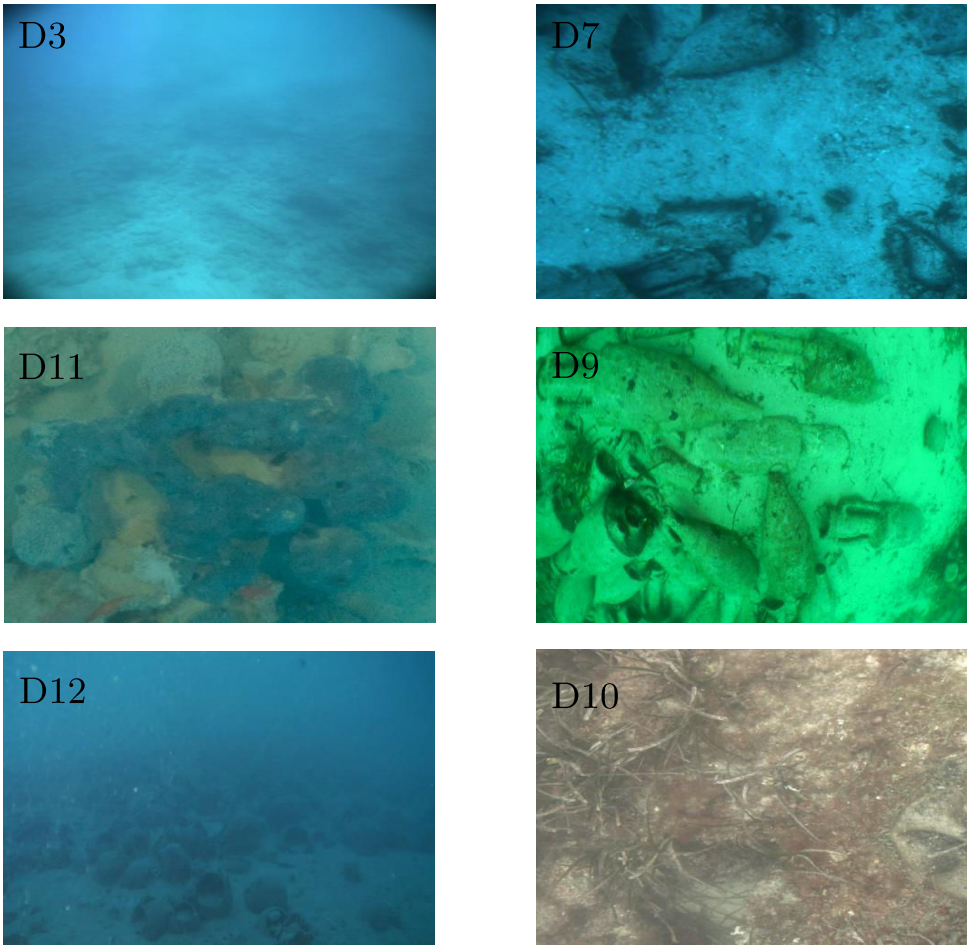


FIGURE 6.13: Examples of image taken from corresponding datasets. It is possible to see that images on column *a)* are characterized by a higher amount of haze than those in column *b)*. These cause different performance of the two feature set used for classification. The uwLBP clearly outperforms the classic LBP with the hazy images as those in the left column.

ally the uwLBP outperforms the classic LBP in highly hazy contexts. In fact the D3, D11 and D12 are exactly those in which the haze is more present meaning that diffused scattering events are well tolerated by these new features and they may lead to better

classify haze-affected underwater images. The chart in Figure 6.14 reports in deep the performance differences obtained in the haziest dataset.

Anyhow it is not discouraged to use uwLBP over clear images, even if in this case the

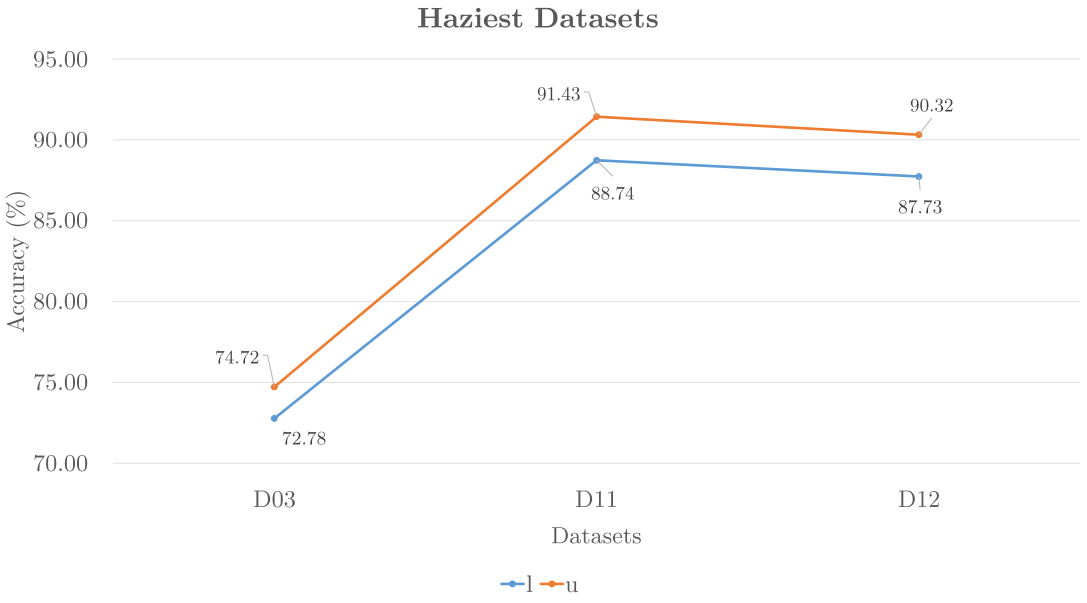


FIGURE 6.14: Overall performance obtained in the haziest datasets with respect to the two employed feature sets.

performance may vary depending more intrinsic image characteristics; sometimes, the classic LBP might represent the best features to use.

In conclusion, the *Underwater LBP* is a feature set derived from the classic LBP and specifically aimed to be used in critical (in the sense of image acquisition) underwater environments. In particular they are focused in dealing with hazy images caused by scattering events due to the presence of suspended particles in the water medium.

Both feature types—LBP and uwLBP—have been tested on our reference datasets, with SVM based classification and related to a variable number of classes. Obtained results showed consistently better performance of uwLBP in almost all the analysed situations; this is a result obtained by averaging all our tests.

While in underwater clearest images the choice between uwLBP and classic LBP might be, as expected, not unique, in dealing with images characterized by an evident presence of distortions—that makes the image to be hazy—the use of uwLBP should be preferred. In fact, results suggest that using this new feature in most difficult underwater images having a high presence of scattering, may lead to obtain higher performance in terms of classification accuracy, differently from the classic LBP.

The uwLBP feature set has been defined with a particular focus on underwater environment. However the haze might be sometimes present also in terrestrial images, in

particular those acquired from satellites. We realized some preliminary (promising) test on this kind of images, but for now this field is left for future extension.

Chapter 7

Conclusions

Computer vision and pattern analysis techniques find a relatively new employment in the field of underwater inspection.
The technology is ready to allow modern underwater robots, in particular AUVs, to be equipped with optical sensor and the related computational hardware to deal with it. Advanced image processing framework may be directly mounted on these vehicles providing the capabilities of real-time analysis and improving the possibility to realize a context-driven navigation (i.e. the chance to take autonomous decision based on what the vehicle actually see) with the final scope of making more accurate and effective seabed inspections. This requires that the image processing and learning algorithms have no delay respect the actual AUV navigation.
Peculiarities introduced by the underwater environment imply that not all the classic algorithms of vision and pattern analysis can be effectively suited. Differences occur both in the type of involved scenario and in the characteristics of the transmission medium. Absorption and scattering are two phenomena strongly related to the light propagation in water and for this reason they need to be properly handled.
Computer vision studies related to underwater scenario are more limited in comparison to the ones regarding the terrestrial environment. Considering all the phenomena linked to the physical aspects of light transmission, underwater computer vision requires more complex models.

This thesis has started with a deep study regarding the light propagation in water medium, analysing its properties and its difficulties in comparison to the air medium and discovering important cues that helped us in solving specific issues.
Almost the entire work was dedicated to the underwater image classification. A complete framework has been proposed to process, principally in real-time, images of the seabed acquired by autonomous underwater vehicles.

After many experiments and evaluations, all the acquired knowledge led us to the definition of a new type of image descriptor specifically designed for underwater environment. The architecture of the developed classification framework macroscopically follows the classic supervised machine learning approach. Hence there are two distinguished phases: one for training and one for the proper image classification. As preferred approach was chosen the SVM (Support vector Machines) due for their achieved performance both in accuracy and required computational time.
The real-time execution constraint has been taken into consideration in all the phases, from image preprocessing, to segmentation and feature extraction. In certain situation we openly sacrificed accuracy performance to stay in reasonable temporal limits.
Thanks to the underwater videos recorded during the ARROWS project, twelve main dataset have been created to deeply evaluate our framework. All these datasets were composed by manually segmenting and labelling thousands of image patches.
The usefulness of collected data, all differentiated by scene type and class, is also meant for an upcoming public release to the vision and pattern analysis community to compensate the current lack of this kind of images, motivate further studies and hopefully to become a shared benchmark for future improvements and works in this field.

Underwater scenarios may present a vivacious variety. The training phase was found crucial for the future performance of the algorithm. For this reason, to achieve a significant underwater classification, several classifiers must be trained depending on the environment and on the classification. We found that training classifiers with images from diverse environmental characteristics may strongly affect the final performance on unseen data, so it is more desirable to use consistent dataset for specifically environment. An a priori classifier selection based on known characteristics of the environment to be explored, seemed to support more efficiently the use of AUV navigation.
For an off-line use with recorded videos our framework instead is able to employ multiple classifiers over a single image with a voting scheme to actually select the most likely class for each processed patch. Alternatively an automatic selection may be a priori conducted based on image appearance characteristics.
The underwater environment does not present sharp and regular shapes and is mostly characterized by planar, irregular and self-similar surfaces.
An extensive evaluation has been conducted with our framework regarding different and time-efficient feature sets. After a preliminary evaluation on textural features, our work deeply addressed to employ first- and second-order statistical features and Local Binary patterns.
A high number of experiments was conducted to evaluate both the single feature set performance over all datasets than under which configurations these features can be used

and have the best behaviour. In fact, other than feature set each single configuration was also related to the input image, number of classes and SVM configuration parameters, such as the kernel function. The obtained results was then compared accordingly to their measured accuracy in the classification stage. Globally the LBP-based descriptor is the one that shows best performances (averaged) over all conducted experiments. All the three descriptors have a similar nature, but we see that using a more structural based approach gives us better results.
Looking at best performance on single experiments, first order statistical features may locally achieve better results, anyhow they are never widely greater and hence there are not cases that totally discourage the use of LBP features. First and second order statistical-based features are inclined to have a higher generalization but they are less capable to correctly discriminate between classes. In addition to this, LBP features are those less affected by the increase of classes that have to be discriminated.
In conclusion the LBP-based descriptor was the one that better get the intrinsic characteristics of the underwater environmental appearance in comparison to other tested features.

Starting from these experiments, in the second part of this work, we also noticed that the image degradation caused by water medium may often noticeably interfere with the feature estimation for what concerns the robustness, reliability and hence the final accuracy.
From the analysis of the physical light underwater transmission we proposed a new kind of features, derived from the LBP and denoted as *Underwater-LBP* (uwLBP). Regarding the classic uniform LBP, a high presence of scattering events associated to many pixels, leads to have an increasing number of non-uniform patterns and consequently causing a general reduction in the discrimination capabilities of this feature set.
Underwater LBPs share the basic approach of classic binary pattern, and they have the same invariance properties. However in comparison of the classic uniform LBP, the proposed underwater version is more reliable in finding uniform patterns in presence of spotted light inconsistencies. In fact, by properly compute values on the reference neighbouring pixels, each time are selected only the ones that theoretically would be less distorted by the underwater environment. In this way the final descriptor is more robust and degradation effects induced by scattering and absorption are controlled.
Both classic LBPs and uwLBPs have been tested on our twelve reference dataset, with SVM based classification and with a variable number of classes. Obtained results confirmed our initial hypothesis by showing a slightly, but consistent, better performance of uwLBPs in almost all the analysed configurations. Although the use of uwLBPs and

classic LBPs might be equally efficient in analysing clearest underwater images, in dealing with hazy images uwLBPs are preferred. Our tests suggest that using this feature in most difficult underwater images with high presence of scattering, may lead to obtain higher performance in terms of final classification accuracy. The achieved results through uwLBPs were 3-4 percentage points better than classic LBP.
Actually the theory behind the definition of this latter Underwater-LBPs is due to the deep investigation of the interaction between the light—that originates images—and water medium. The sea water is a colloidal system that has light transmission properties significantly lower in comparison of the air. The main phenomena to be taken into consideration are related to the absorption and mostly the scattering of electromagnetic waves.
The effect of scattering on acquired (terrestrial) images has been previously dealt in several works, but only recently have been proposed approaches capable to recover the actual image radiance directly from a single image and without the use of additional hardware. In this thesis we started with images from the terrestrial scenario, with a deep review of the most important works on this topic in particular focusing the attention on those based on DCP (Dark Channel Prior). Our implementation, with some slight changes, showed comparable results with the original work.
Compatibly to the main objective of this work, we then tested the terrestrial approaches on underwater environment, but the obtained results were definitively poorer.
The main issues were principally related to the presence of evident absorption light effects and of a non-uniform, often artificial, illumination. On these basis we developed a new dehazing method, based on physical model of light transmission in water especially directed to the underwater scenario. There are very few works about the underwater dehazing that are not directly a re-proposal of already adopted terrestrial approaches. Our proposed method is the only one that actually makes an adaptive airlight estimation other than adopting a different new model for the transmission evaluation. From a comparative qualitative evaluation in all our dataset we saw that our new method achieve qualitative results better than the other main existing approaches for what concerns the recovered image details and the global illumination.

In conclusion this thesis has carried out three macroscopic contributions all related to the theme of applying computer vision and pattern recognition techniques to the underwater environment. The deep study of existing feature sets that can be effectively employed to classify the seabed; the study of terrestrial dehazing techniques and the development of a new way to improve the actual radiance recovery in underwater images; the development of a new LBP-based feature set, for classification in poor visibility underwater environment.

All this work was surrounded by the realization of a software framework for seabed inspection that can be also actually installed on an AUV to automatically classify images, by allowing a context driven navigation.

Future works can be addressed to various directions. The underwater environment is still—in our opinion and considering the number of related works—poorly studied by the computer vision researchers and so unsolved issues might be present. Underwater environment is also resulted more challenging than the terrestrial one. Anyhow there are three main big future developments that we want to achieve.
The first is that the feature sets that we deeply investigated can be obviously extended, especially relaxing the temporal constraints and extending the number of datasets. The set of classes that we investigated can also be augmented in the same way and a multi-layer classification can increase the taxonomy levels. The second is related to the dehazing techniques both in water than in air. In fact the obtained transmission map—for now a sub-product of the process of haze removal—can have multiple uses, as discussed at the end of Chapter 3. The coarse 3D extraction and the forensic field, are only two of many employment that is our aim to deeply investigate. Finally, the third and last (macro)direction come out from the consideration that, other than further extensive tests and comparisons, the developed underwater-LBPs can be also applied to terrestrial images taken in bad atmospheric condition. In fact, we hypothesize—with marginal changes—a good performance of uwLBPs on air as well as in water.

Bibliography

[1] S. Bazeille, I. Quidu, and J.P. Jaulin, L.and Malkasse. Automatic underwater image pre-processing. *Proceedings of CMM 2006*, 1900:8, 2006.

[2] R. T. Tan. Visibility in bad weather from a single image. *26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2008.

[3] K. Nishino, L. Kratz, and S. Lombardi. Bayesian defogging. *International Journal of Computer Vision*, 98(3):263–278, 2012.

[4] K. He, J. Sun, and X. Tang. Single image haze removal using dark channel prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(12):2341–2353, 2011.

[5] R. Fattal. Single image dehazing. *ACM Transactions on Graphics*, 27(3):1, 2008.

[6] H. Wen, Y. Tian, T. Huang, and W. Gao. Single underwater image enhancement with a new optical model. *Proceedings - IEEE International Symposium on Circuits and Systems*, pages 753–756, 2013.

[7] H. Hirschmuller and D. Scharstein. Evaluation of Cost Functions for Stereo Matching. *Proc. of CVPR*, pages 1–8, 2007.

[8] D. Scharstein and C. Pal. Learning Conditional Random Fields for Stereo. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[9] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

[10] Z. Guo, L. Zhang, and D. Zhang. Rotation invariant texture classification using LBP variance (LBPV) with global matching. 43(3):706–719, 2010.

[11] M. Sokolova and G. Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing and Management*, 45(4):427–437, 2009.

[12] S. Corchs and R. Schettini. Underwater image processing: State of the art of restoration and image enhancement methods. *Eurasip Journal on Advances in Signal Processing*, 2010, 2010.

[13] Eugene Hecht. *Optics (4th Ed.)*. Addison-Wesley, 2001.

[14] T.J. Petzold. Volume Scattering Functions for Selected Ocean Waters. *Scripps Inst. Oceanogr.*, (October):72–78, 1972.

[15] A. Quirantes and S. Bernard. Light scattering by marine algae: Two-layer spherical and nonspherical models. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 89(1-4):311–321, 2004.

[16] S. Q. Duntley. Light in the Sea. *Journal of the Optical Society of America*, 53: 214–233, 1963.

[17] A. Morel. Optical properties of pure water and pure sea water. *optical Aspects of Oceanograph*, pages 1–24, 1974.

[18] B. McGlamery. Computer analysis and simulation of underwater camera system performance. *SIO ref*, 1975.

[19] J. S. Jaffe. Computer modeling and the design of optimal underwater imaging systems. *Oceanic Engineering, IEEE Journal of*, 15(2):101–111, 1990.

[20] M. L. Wells and E. D. Goldberg. Colloid aggregation in seawater. *Marine Chemistry*, 41:353–358, 1993.

[21] A. Morel and A. Bricaud. Theoretical results concerning light absorption in a discrete medium, and application to specific absorption of phytoplankton. *Deep Sea Research Part A. Oceanographic Research Papers*, 8:1375–1393, 1981.

[22] R. Li, H. Li, W. Zou, R. G. Smith, and T. A. Curran. Quantitative photogrammetric analysis of digital underwater video imagery. *IEEE Journal of Oceanic Engineering*, 22(2):364–375, 1997.

[23] A. Sedlazeck, K. Koser, and R. Koch. 3D reconstruction based on underwater video from ROV Kiel 6000 considering underwater imaging conditions. *Oceans 2009-Europe*, pages 1–10, May 2009.

[24] J. M. Lavest, G. Rives, and J. T. Lapresté. Underwater Camera Calibration. *Lecture Notes in Computer Science*, 1843:654–668, 2000.

[25] A. Sedlazeck and R. Koch. Simulating Deep Sea Underwater Images Using Physical Models for Light Attenuation, Scattering, and Refraction. *Proceedings of the Vision, Modeling, and Visualization Workshop*, pages 49–56, 2011.

[26] W. Hou, D. J. Gray, A. D. Weidemann, and R. A. Arnone. Comparison and validation of point spread models for imaging in natural waters. *Optics express*, 16(13):9958–9965, 2008.

[27] E. Trucco and A. T. Olmos-Antillon. Self-tuning underwater image restoration. *IEEE Journal of Oceanic Engineering*, 31(2):511–519, 2006.

[28] Y. Y. Schechner and N. Karpel. Recovery of underwater visibility and structure by polarization analysis. *IEEE Journal of Oceanic Engineering*, 30(3):570–587, 2005.

[29] T. Treibitz and Y. Y. Schechner. Active polarization descattering. *IEEE transactions on pattern analysis and machine intelligence*, 31(3):385–99, 2009.

[30] A. Arnold-Bos, J.P. Malkasse, and G. Kervern. A preprocessing framework for automatic underwater images denoising. *European Conference on Propagation and Systems*, (1):8, 2005.

[31] M. Chambah, D. Semani, A. Renouf, P. Courtellemont, and A. Rizzi. Underwater color constancy: enhancement of automatic live fish recognition. *Proceedings of SPIE*, 5293:157–168, 2003.

[32] L. Torres-Méndez and G. Dudek. Color correction of underwater images for aquatic robot inspection. *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 60–73, 2005.

[33] R. Garcia, T. Nicosevici, and X. Cufi. On the way to solve lighting problems in underwater imaging. *Oceans '02 Mts/Ieee*, 2(1):1018–1024, 2002.

[34] F. Petit, A.S. Capelle-Laizé, and P. Carré. Underwater image enhancement by attenuation inversion with quaternions. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, (November 2015):1177–1180, 2009.

[35] A. T. Çelebi and S. Ertürk. Visual enhancement of underwater images using Empirical Mode Decomposition. *Expert Systems with Applications*, 39(1):800–805, 2012.

[36] T. Ji and G. Wang. An approach to underwater image enhancement based on image structural decomposition. *Journal of Ocean University of China*, 14(2): 255–260, 2015.

[37] C. O. Ancuti, C. Ancuti, C. Hermans, and P. Bekaert. Enhancing Underwater Images And Videos By Fusion. *Lecture Notes in Computer Science*, 6493 LNCS: 501–514, 2011.

[38] M. Sheng, Y. Pang, L. Wan, and H. Huang. Underwater Images Enhancement Using Multi-Wavelet Transform and Median Filter. 12(3):2306–2313, 2014.

[39] S. Wong, Y. Yu, N. A. Ho, and R. Paramesran. Comparative analysis of underwater image enhancement methods in different color spaces. *2014 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, pages 034–038, 2014.

[40] M. Arredondo and K. Lebart. A methodology for the systematic assessment of underwater video processing algorithms. *Europe Oceans 2005*, 1:362–367, 2005.

[41] W. Hou and A. D. Weidemann. Objectively assessing underwater image quality for the purpose of automated restoration. *Proceedings of SPIE*, 6575(0704):65750Q–65750Q–7, 2007.

[42] Z. Wang, A. C. Bovik, and E. P. Sheikh, H. R.and Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[43] M. S. Bewley, B. Douillard, N. Nourani-Vatani, a. Friedman, O. Pizarro, and S. B. Williams. Automated species detection: An experimental approach to kelp detection from sea-floor AUV images. *Australasian Conference on Robotics and Automation, ACRA*, 2012.

[44] G. A. Hollinger, U. Mitra, and G. S. Sukhatme. Active Classification: Theory and Application to Underwater Inspection. *arXiv:1106.5829 [cs.RO]*, page 16, 2011.

[45] J.J. Leonard, A. A. Bennett, C.M. Smith, H. Jacob, and S. Feder. Autonomous underwater vehicle navigation. In *MIT Marine Robotics Laboratory Technical Memorandum*, 1998.

[46] J. Jaffe, K. Moore, J. McLean, and M. Strand. Underwater Optical Imaging: Status and Prospects. *Oceanography*, 14(3):64–75, 2001.

[47] D. M. Kocak, F.R. Dalgleish, F.M. Caimi, and Y.Y. Schenchner. A focus on recent developments and trends in underwater imaging. *Marine Technology Society Journal*, 42(November 2003), 2008.

[48] M. A. Fairweather and A. R. Hodgetts. Robust scene interpretation of underwater image sequences. In *Proc. on Image Processing and Its Applications*, pages 660–664, 1997.

[49] A. Olmos, M. Trucco, K. Lebart, and D. Lane. Detecting ripple patterns in mission videos. *OCEANS 2000 MTS/IEEE Conference and Exhibition. Conference Proceedings (Cat. No.00CH37158)*, 1, 2000.

[50] K. Lebart, E. Trucco, and D. M. Lane. Real-time automatic sea-floor change detection from video. In *OCEANS 2000 MTS/IEEE Conference and Exhibition.*, number 111, pages 1337–1343, 2000.

[51] K. Lebart, C. Smith, E. Trucco, and D. M. Lane. Automatic indexing of underwater survey video: Algorithm and benchmarking method. *IEEE Journal of Oceanic Engineering*, 28(4):673–686, 2003.

[52] D. Walther, D.R. Edgington, and C. Koch. Detection and tracking of objects in underwater video. *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, 1:0–5, 2004.

[53] S. Bazeille, I. Quidu, and L. Jaulin. Identification of underwater man-made object using a colour criterion. *Classification of Underwater*, 29:45–52, 2007.

[54] J. Hu, H. Zhang, A. Miliou, T. Tsimpidis, H. Thornton, and V. Pavlovic. Categorization of underwater habitats using dynamic video textures. *Proceedings of the IEEE International Conference on Computer Vision (Workshop)*, pages 838–843, 2013.

[55] M. Dunbabin and L. Marques. Robots for environmental monitoring: Significant advancements and applications. *IEEE Robotics and Automation Magazine*, 19(1): 24–39, 2012.

[56] ARROWS. Archaeological robot systems for the world's seas, 2012. URL `http://www.arrowsproject.eu/`.

[57] THESAURUS. Tecniche per l'esplorazione sottomarina archeologica mediante l'utilizzo di robot autonomi in sciami, 2012. URL `http://thesaurus.isti.cnr.it/`.

[58] Allotta Benedetto, Costanzi Riccardo, Ridolfi Alessandro, Colombo Carlo, Bellavia Fabio, Fanfani Marco, Pazzaglia Fabio, Salvetti Ovidio, Moroni Davide, Pascali Maria Antonietta, Reggiannini Marco, Kruusmaa Maarja, Salum ae Taavi, Frost Gordon, Tsiogkas Nikolaos, Lane David M, Cocco Michele, Gualdesi Lavinio, Roig Daniel, G undogdu Hilal Tolasa, Tekdemir Enis I, Dede Mehmet Ismet Can, Baines Steven, Agneto Floriana, Selvaggio Pietro, Tusa Sebastiano, Zangara Stefano, Dresen Urmas, L atti Priit, Saar Teele, and Daviddi Walter. The arrows project: adapting and developing robotics technologies for underwater archaeology. *IFAC - International Federation of Automatic Control*, 2015.

[59] Allotta Benedetto, Baines Steven, Bartolini Fabio, Bellavia Fabio, Colombo Carlo, Conti Roberto, Costanzi Riccardo, Dede Can, Fanfani Marco, Gelli Jonathan, Gündogdu Hilal Tolasa, Monni Niccolò, Moroni Davide, Natalini Marco, Pascali Maria Antonietta, Pazzaglia Fabio, Pugi Luca, Ridolfi Alessandro, Reggiannini Marco, Roig Daniel, Salvetti Ovidio, and Tekdemir Enis. Design of a modular autonomous underwater vehicle for archaeological investigations. In *OCEANS'15 MTS/IEEE GENOVA*. IEEE, 2015.

[60] A. Bosch, A. Zisserman, and X. Munoz. Representing shape with a spatial pyramid kernel. 2007.

[61] A. Kumar and C. Sminchisescu. Support Kernel Machines for Object Recognition. In *ICCV 2007. IEEE 11th International Conference on*, 2007.

[62] Y. Lin, T. Liu, and C. Fuh. Local Ensemble Kernel Learning for Object Category Recognition. *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, 2007.

[63] N. García-Pedrajas and D. Ortiz-Boyer. An empirical study of binary classifier fusion methods for multiclass classification. *Information Fusion*, 12(2):111–130, 2011.

[64] P. S. Chavez. An improved dark-object subtraction technique for atmospheric scattering correction of multispectral data. *Remote Sensing of Environment*, 24 (3):459–479, 1988.

[65] J.P. Oakley and B.L. Satherley. Improving image quality in poor visibility conditions using a physical model for contrast degradation. *Image Processing, IEEE Transactions on*, 7:70, 1998.

[66] A. Levin, R. Fergus, F. Durand, and W. T. Freeman. Image and depth from a conventional camera with a coded aperture. *ACM Transactions on Graphics*, 26 (3):70, 2007.

[67] S. Shwartz, E. Namer, and Y. Y. Schechner. Blind haze separation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2:1984–1991, 2006.

[68] F. Cozman and E. Krotkov. Depth from scattering. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1997.

[69] S. K. Nayar and S. G. Narasimhan. Vision in bad weather. *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 2(c):820–827, 1999.

[70] C. F. Bohren. *Handbook of Optics*, volume 1 of *3*. McGraw-Hill Education, 2009. Chapter 7. Scattering by Particles.

[71] J. Wang, N. He, L. Zhang, and K. Lu. Single image dehazing with a physical model and dark channel prior. *Neurocomputing*, 149:718–728, 2015.

[72] M. Ding and R. F. Tong. Efficient dark channel based image dehazing using quadtrees. *Science China Information Sciences*, 56(9):1–9, 2013.

[73] M. Toda, K. Senzaki, and M. Tsukada. Image Clarification Method Based on Structure-Texture Decomposition with Texture Refinement. In *18th International Conference on Image Analysis and Processing (ICIAP)*, 2015.

[74] C. Hsieh, Y. Lin, and C. Chang. Haze removal without transmission map refinement based on dual dark channels. In *Machine Learning and Cybernetics (ICMLC), 2014 International Conference on*, pages 13–16, 2014.

[75] X. Lv, W. Chen, and I. F. Shen. Real-time dehazing for image and video. *Proceedings - Pacific Conference on Computer Graphics and Applications*, pages 62–69, 2010.

[76] J. H. Kim, W. D. Jang, J. Y. Sim, and C. S. Kim. Optimized contrast enhancement for real-time image and video dehazing. *Journal of Visual Communication and Image Representation*, 24(3):410–425, 2013.

[77] P. Carr and R. Hartley. Improved Single Image Dehazing Using Geometry. *2009 Digital Image Computing: Techniques and Applications*, pages 103–110, 2009.

[78] C. Liu, W. T. Freeman, R. Szeliski, and S. B. Kang. Noise estimation from a single Image. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1:901–908, 2006.

[79] G. Larson, E. Orlando, H. Rushmeier, and C. Piatko. A Visibility Matching Tone Reproduction Operator for High Dynamic Range Scenes. In *Visualization and Computer Graphics, IEEE Transactions on*, volume 3, pages 291 – 306, 1997.

[80] Y. Du, B. Guindon, and J. Cihlar. Haze detection and removal in high resolution satellite image with wavelet analysis. *IEEE Transactions on Geoscience and Remote Sensing*, 40(1):210–217, 2002.

[81] J. Zhang, L. Li, G. Yang, Y. Zhang, and J. Sun. Local albedo-insensitive single image dehazing. *Visual Computer*, 26(6-8):761–768, 2010.

[82] N. Hautière, J. P. Tarel, D. Aubert, and E. Dumont. Blind contrast enhancement assessment by gradient ratioing at visible edges. *Image Analysis and Stereology*, 27(2):87–95, 2008.

[83] D. Wu, Q. Zhu, J. Wang, Y. Xie, and L. Wang. Image Haze Removal: Status, Challenges and Prospects. In *Information Science and Technology (ICIST)*, pages 492 – 497, 2014.

[84] Y. Y Schechner, S. G. Narasimhan, and S. K. Nayar. Polarization-based vision through haze. *Applied optics*, 42(3):511–525, 2003.

[85] S. G. Narasimhan and S. K. Nayar. Interactive (de) weathering of an image using physical models. *IEEE Workshop on Color*, pages 1–8, 2003.

[86] K. Wang, E. Dunn, J. Tighe, and J.M. Frahm. Combining semantic scene priors and haze removal for single image depth estimation. *2014 IEEE Winter Conference on Applications of Computer Vision, WACV 2014*, pages 800–807, 2014.

[87] L. Yuan, J. Sun, L. Quan, and H. Shum. Image deblurring with blurred/noisy image pairs. *ACM Transactions on Graphics*, 26(3):1, 2007.

[88] S. G. Narasimhan and S. K. Nayar. Vision and the atmosphere. *International Journal of Computer Vision*, 48(3):233–254, 2002.

[89] A. Levin, D. Lischinski, and Y. Weiss. A closed-form solution to natural image matting (Short). *Pami*, 30(2):228–42, 2008.

[90] L. Chao and M. Wang. Removal of water scattering. *ICCET 2010 - 2010 International Conference on Computer Engineering and Technology, Proceedings*, 2: 35–39, 2010.

[91] Z. Chen, H. Wang, J. Shen, X. Li, and L. Xu. Region-specialized underwater image restoration in inhomogeneous optical environments. *Optik*, 125(9):2090–2098, 2014.

[92] J.Y. Chiang and Y. C. Chen. Underwater image enhancement by wavelength compensation and dehazing. *IEEE Transactions on Image Processing*, 21(4):1756–1769, 2012.

[93] N. Carlevaris-Bianco, A. Mohan, and R. M. Eustice. Initial results in underwater single image dehazing. *MTS/IEEE Seattle, OCEANS 2010*, 2010.

[94] P. Drews-Jr, E. Do Nascimento, F. Moraes, S. Botelho, and M. Campos. Transmission estimation in underwater single images. *Proceedings of the IEEE International Conference on Computer Vision*, (1):825–830, 2013.

[95] K. He, J. Sun, and X. Tang. Guided image filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(6):1397–1409, 2013.

[96] K. He and J. Sun. Fast Guided Filter. *arXiv*, pages 2–3, 2015.

[97] C. Tomasi and R. Manduchi. Bilateral Filtering for Gray and Color Images. *International Conference on Computer Vision*, pages 839–846, 1998.

[98] M. Sulami, I. Geltzer, R. Fattal, and M. Werman. Automatic recovery of the atmospheric light in hazy images. In *IEEE International Conference on Computational Photography (ICCP)*, 2014.

[99] Z. Wang, H.R. Sheikh, and a.C. Bovik. No-reference perceptual quality assessment of JPEG compressed images. *Proceedings. International Conference on Image Processing*, 1:477–480, 2002.

[100] L. Zhao, M. Hansard, and A. Cavallaro. Pop-up Modelling of Hazy Scenes. In *Lecture Notes in Computer Science*, pages 306–318, 2015.

[101] I. Bülthoff, H. Bülthoff, and P. Sinha. Top-down influences on stereoscopic depth-perception. *Nature neuroscience*, 1(3):254–257, 1998.

[102] J. Malik and R. Rosenholtz. Computing Local Surface Orientation and Shape from Texture for Curved Surfaces. *Ijcv*, 23(2):149–168, 1997.

[103] R.Z.R. Zhang, P.S. Tsai, J.E. Cryer, and M. Shah. Shape-from-shading: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8):690–706, 1999.

[104] E. Delage and A. Y. Ng. A Dynamic Bayesian Network Model for Autonomous 3D Reconstruction from a Single Indoor Image. *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Volume 2 CVPR06*, 2: 2418–2428, 2006.

[105] A. Saxena, J. Schulte, and A. Y. Ng. Depth Estimation using Monocular and Stereo Cues. *Proc. 20th Int. Joint Conf. Artificial Intelligence*, (August 2007): 2197–2203, 2007.

[106] A. Saxena, M.S.M. Sun, and A.Y. Ng. Make3D: Learning 3D Scene Structure from a Single Still Image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5), 2009.

[107] H. Farid. Image forgery detection. *IEEE Signal Processing Magazine*, 2009.

[108] K. Karu, A. K. Jain, and R. M. Bolle. Is there any texture in the image? *Pattern Recognition*, 29(9):1437–1446, 1996.

[109] L. W. Renninger and J. Malik. When is scene identification just texture recognition? *Vision Research*, 44(19):2301–2311, 2004.

[110] M. Varma and A. Zisserman. A statistical approach to texture classification from single images. *International Journal of Computer Vision*, 62(1-2):61–81, 2005.

[111] O. Beijbom, P. J. Edmunds, D. I. Kline, B. G. Mitchell, and D. Kriegman. Automated annotation of coral reef survey images. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1170–1177, 2012.

[112] W. Su, J. Li, Y. Chen, Z. Liu, J. Zhang, T. M. Low, I. Suppiah, and S.A.M. Hashim. Textural and local spatial statistics for the object-oriented classification of urban areas using high resolution imagery. *International Journal of Remote Sensing*, 29(11):3105–3117, 2008.

[113] A. Baraldi and F. Parmiggiani. An investigation of the textural characteristics associated with gray level cooccurrence matrix statistical parameters. *IEEE Transactions on Geoscience and Remote Sensing*, 33(2):293–304, 1995.

[114] S. Arivazhagan and L. Ganesan. Texture classification using wavelet transform. *Pattern Recognition Letters*, 24:1513–1521, 2003.

[115] J. Mashford, M. Rahilly, and D. Marney. Processing by SVM of Haar Wavelet Transforms for Discontinuity Detection. In *Proceedings of The 2011 World Congress in Computer Science, Computer Engineering, and Applied Computing*, 2011.

[116] L. Zhang, W. Zhou, and L. Jiao. Wavelet Support Vector Machine. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 34(1):34–39, 2004.

[117] N. Aggarwal and R. K. Agrawal. First and Second Order Statistics Features for Classification of Magnetic Resonance Brain Images. *Journal of Signal and Information Processing*, 3(2):146–153, 2012.

[118] J. Shotton, J. Winn, C. Rother, and A. Criminisi. {TextonBoost} for Image Understanding: Multi-Class Object Recognition and Segmentation by Jointly Modeling Appearance, Shape and Context. *Int. Journal of Computer Vision*, 81(1):2–23, 2009.

[119] C. Barat and R. Phlypo. A fully automated method to detect and segment a manufactured object in an underwater color image. *Eurasip Journal on Advances in Signal Processing*, 2010.

[120] J.M.H. Du Buf, M. Kardan, and M. Spann. Texture feature performance for image segmentation. *Pattern Recognition*, 23:291–309, 1990.

[121] R. P. Nikhil and K. P. Sankar. A review on image segmentation techniques. *Pattern Recognition*, 26:1277–1294, 1993.

[122] J. Smith and S. F. Chang. Quad-tree segmentation for texture-based image query. *Proceedings of the second ACM international conference on Multimedia*, pages 279–286, 1994.

[123] B.S. Manjunath and R. Chelappa. Unsupervised texture segmentation using markov random field models. *Pattern Analysis and Machine Intelligence*, 13:478–482, 1991.

[124] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2281, 2012.

[125] A. Materka and M. Strzelecki. Texture Analysis Methods – A Review. 11:1–33, 1998.

[126] L. Van Gool, P. Dewaele, and A. Oosterlinck. Texture analysis. *Computer Vision, Graphics, and Image Processing*, 29:336–357, 1985.

[127] J. Zhang and T. Tan. Brief review of invariant texture analysis methods. *Pattern Recognition*, 35:2301–2311, 2002.

[128] R. M. Haralick. Statistical and Structural Approaches To Texture. *Proc IEEE*, 67 (5):786–804, 1979.

[129] R. L. Kashyap and A. Khotanzad. A Model-Based Method for Rotation Invariant Texture Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(4):472–481, 1986.

[130] F.S. Cohen, Z. Fan, and M.A. Patel. Classification of Rotated and Scaled Textured Images Using Gaussian Markov Random Field Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:192–202, 1991.

[131] T. Chang and C. J. Kuo. Texture analysis and classification with tree-structured wavelet transform. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, 2(4):429–441, 1993.

[132] P. Pentland. Fractal-based description of natural scenes. *IEEE transactions on pattern analysis and machine intelligence*, 6(6):661–74, June 1984.

[133] K. Jain and F. Farrokhnia. Unsupervised texture segmentation using Gabor filters. *1990 IEEE International Conference on Systems Man and Cybernetics Conference Proceedings*, 000(12):14–19, 1990.

[134] N. Pican, E. Trucco, M. Ross, D.M. Lane, Y. Petillot, and I. T. Ruiz. Texture analysis for seabed classification: co-occurrence matrices vs. self-organizing maps. *IEEE Oceanic Engineering Society. OCEANS'98. Conference Proceedings (Cat. No.98CH36259)*, 1:424–428, 1998.

[135] C. C. Gotlieb and H. E. Kreyszig. Texture descriptors based on co-occurrence matrices. *Computer Vision, Graphics, and Image Processing*, 51:70–86, 1990.

[136] T. Kohonen and T. Kohonen. The self-organizing map. *Neurocomputing*, 21, 1998.

[137] R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural Features for Image Classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 3(6), 1973.

[138] D.A. Clausi. An analysis of co-occurrence texture statistics as a function of grey level quantization. *Canadian Journal of Remote Sensing*, 28(1):45–62, 2002.

[139] L. Wang and D.C. He. Texture classification using texture spectrum. *Pattern Recognition*, 23:905–910, 1990.

[140] R. Zabih and J. Woodfill. Non-parametric Local Transforms for Computing Visual Correspondence. *In Proceedings of European Conference on Computer Vision*, (May):151–158, 1994.

[141] T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29 (1):51–59, 1996.

[142] M. Pietikäinen, T. Ojala, and Z. Xu. Rotation-invariant texture classification using feature distributions. *Pattern Recognition*, 33(1):43–52, 2000.

[143] T. Ojala, K. Valkealahti, E. Oja, and M. Pietikäinen. Texture discrimination with multidimensional distributions of signed gray-level differences. *Pattern Recognition*, 34(3):727–739, 2001.

[144] J. Trefný and J. Matas. Extended set of local binary patterns for rapid object detection. *Computer Vision Winter Workshop*, pages 1–7, 2010.

[145] G. Zhao and M. Pietikainen. Local Binary Pattern Descriptors for Dynamic Texture Recognition. *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, pages 0–3, 2006.

[146] G. Zhao and T. Ahonen. Rotation Invariant Image and Video Description with Local Binary Pattern Features. pages 1–13, 2010.

[147] A. Torralba and A. A. Efros. Unbiased look at dataset bias. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1521–1528, 2011.

[148] N.S. Barrett, L. Meyer, N. Hill, and P.H. Walsh. Methods for the processing and scoring of AUV digital imagery from South Eastern Tasmania. pages 0–51, 2011.

[149] M.S. Bewley, B. Douillard, N. Nourani-Vatani, A.L. Friedman, O. Pizarro, and S.B. Williams. Automated species detection: An experimental approach to kelp. *Proceedings Australasian Conference on Robotics and Automation (ACRA) 2012*, 2012.

[150] D. Ruderman. The statistics of natural images. *Network: Computation in Neural Systems*, 5(4):517–548, 1994.

[151] A. Torralba and A. Oliva. Statistics of natural image categories. *Network (Bristol, England)*, 14(3):391–412, 2003.

[152] J. J. Koenderink. The structure of locally orderless images. *International Journal of Computer Vision*, 31:159–168, 1999.

[153] M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[154] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2: 2169–2178, 2006.

[155] S. Liao, X. Zhu, Z. Lei, L. Zhang, and S. Li. Learning Multi-scale Block Local Binary Patterns for Face Recognition. *Advances in Biometrics*, pages 828–837, 2007.

[156] O. Boiman, E. Shechtman, and M. Irani. In defense of Nearest-Neighbor based image classification. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[157] X. Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.

[158] C.J.C. Christopher J. C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[159] V. N. Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, 10(5):988–999, 1999.

[160] N. Cristianini and J. Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

[161] J. D. M. Rennie, L. Shih, J. Teevan, and D. R. Karger. Tackling the Poor Assumptions of Naive Bayes Text Classifiers. *Proceedings of the Twentieth International Conference on Machine Learning (ICML)-2003)*, 20(1973):616–623, 2003.

[162] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. *Proceedings of the ECCV International Workshop on Statistical Learning in Computer Vision*, pages 59–74, 2004.

[163] C. Chang and C. Lin. LIBSVM : A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2:1–39, 2011.

[164] V. Jumutc and J. Suykens. Multi-Class Supervised Novelty Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1–1, 2014.

[165] C. Hsu and C. Lin. A Comparison of Methods for Multi-class Support Vector Machines. *Neural Networks, IEEE Transactions on*, 13(2):415–425, 2002.

[166] Y. LeCun, L. Jackel, L. Bottou, A. Brunot, C. Cortes, J. Denker, H. Drucker, I. Guyon, U. Müller, E. Säckinger, P. Simard, and V. Vapnik. Comparison of learning algorithms for handwritten digit recognition. *International Conference on artificial neural networks*, pages 53–60, 1995.

[167] S. Knerr, L. Personnaz, and G. Dreyfus. Single-layer learning revisited: a stepwise procedure for building and training a neural network. In J. Fogelman, editor, *Neurocomputing: Algorithms, Architectures and Applications*. Springer-Verlag, 1990.

[168] J. Friedman. Another approach to polychotomous classifcation. *Technical report Dept. Statist. Stanford Univ., Stanford*, 1996.

[169] M. Sokolova, N. Japkowicz, and S. Szpakowicz. Beyond accuracy, F-Score and ROC: A family of discriminant measures for performance evaluation. *Advances in Artificial Intelligence (Lecture Notes in Computer Science)*, 4304(c):1015–1021, 2006.

[170] N. Lachiche and P. Flach. Improving Accuracy and Cost of Two-Class and Multi-Class Probabilistic Classifiers Using ROC Curves. *Proceedings of the Twentieth International Conference on Machine Learning*, pages 416–424, 2003.

[171] T. Ahonen and M. Pietikainen. Soft Histograms For local Binary Patterns. In *Finnish Signal Processing Symposium*, 2007.

[172] J. Zhang and T. Tan. Brief review of invariant texture analysis methods. *Pattern Recognition*, 35(3):735–747, 2002.

[173] Trygve Randen and J. H. Husø y. Filtering for texture classification: A comparative study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(4): 291–310, 1999.

[174] Y.Y. Schechner and N. Karpel. Clear underwater vision. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2004.*, pages 536–543, 2004.

[175] A. Barcelo, E. Montseny, and P. Sobrevilla. Fuzzy Texture Unit and Fuzzy Texture Spectrum for texture characterization. *Fuzzy Sets and Systems*, 158(3):239–252, 2007.

[176] A.K. Jain, R.P.W. Duin, and J. Mao. Statistical pattern recognition: A review. *IEEE transactions on pattern analysis and machine intelligence*, 22(1):4–37, 2000.