Masaryk University
Faculty of Informatics

Localization and Tracking for Underwater Augmented Reality

Ph.D. Thesis Proposal

Mgr. Jan Čejka

Brno, Fall 2018

Masaryk University
Faculty of Informatics

Localization and Tracking for Underwater Augmented Reality

Ph.D. Thesis Proposal

Mgr. Jan Čejka

Advisor: doc. Fotis Liarokapis, Ph.D.

Brno, Fall 2018

Signature of Thesis Advisor

Contents

Introduction **1**

1 State of the art **3**
1.1 *Marker-based detection of objects* 3
1.2 *Marker-less detection of objects* 6
1.3 *Improving images taken in reduced visibility conditions* . . . 8
1.3.1 Improving underwater images 10
1.4 *Tracking and SLAM* 12
1.5 *Underwater augmented reality* 13

2 Aims of the thesis **15**
2.1 *Contributions and expected publications* 15
2.2 *Study plan* 18

3 Achieved results **19**
3.1 *Improving underwater images for detection of markers* . . . 19
3.2 *Detection of markers under water* 21
3.3 *Underwater localization and tracking system* 23
3.4 *Publications* 24

Appendix **27**

Bibliography **29**

Introduction

Historical objects and places play important roles in lives of people, since history is one of the bases of every culture. Old buildings are repaired, protected, and preserved for future generations, and archaeologists search for artifacts to recover data about our past. Sites of archaeological significance are not only on land, but they are located also under water, for example ships that sank into water, or cities that submerged due to earthquakes or tsunami waves. Their remains tell us more about the culture of the sailors and about the history of the sites. During the past decade, people started diving to experience an underwater view of these sites in such high numbers that old ships are intentionally sunk to become artificial shipwrecks for tourists to protect natural underwater sites [1].

Nowadays, technology has evolved and provides new options and opportunities to present the past to people. Advances in the field of mobile devices allow us to use them for augmented reality (AR) applications to see in real time what cannot be seen in the real world, and through a screen of smart phones or tablets, it is possible to visualize in three-dimensions how buildings looked hundreds or thousand years ago [2, 3].

One of the most crucial things that are necessary for AR applications is to know the precise pose (position and orientation) of the device and all important objects around. In urban environments, there are lots of available solutions such as sensors (GPS, Wi-Fi, etc.), or computer vision solutions. Unfortunately, the situation is different under water. In the sea, there is no GPS signal or other signals to get the location of the device, and also due to worsen visibility conditions, objects that are away further than a dozen of meters cannot be distinguished. For these reasons, it is still very hard to recognize objects under water and track their position, so it is necessary to populate the underwater environment with either sensors (e.g., acoustic beacons) or recognizable objects (e.g., markers) to give divers a robust solution to see the history of underwater places in AR.

This research is done as a part of the i-MareCulture project[1] (Advanced VR, iMmersive Serious Games and Augmented REality as

1. https://imareculture.eu/

Tools to Raise Awareness and Access to European Underwater CULTURal heritagE, Digital Heritage), a part that implements an AR solution for divers to provide viable option to see the history of underwater cultural heritage sites. It is focused on improving the performance of computer vision algorithms in harsh underwater sea conditions, detection of objects underwater, and localization and tracking of mobile devices under water.

1 State of the art

State of the art related to my research is divided into the following sections: detection of objects based on fiducial markers (Section 1.1), detection of objects based on natural features without using artificial markers (Section 1.2), techniques for improving images taken in reduced visibility conditions (Section 1.3), tracking and simultaneous localization and mapping techniques (Section 1.4), and solutions for underwater AR (Section 1.5).

1.1 Marker-based detection of objects

Detection of objects using fiducial markers is one of the easiest methods of detection. Markers are artificial objects that are designed for the purpose to be easily recognized and identified; fiducial markers are markers that are at known positions or of known size, so they also serve as reference points or a measure for other objects.

The process of detection and identification of markers is described in Figure 1.1. First, the input image is simplified for further processing, often becoming a binary image by converting the input to a single-channel image (if the input is a multi-channel image, e.g., RGB image) and thresholding it. This simplified image is then searched for candidate markers, which are objects of the same shape as the markers, but it is not yet known whether they are actual markers or other objects. In the next step, each candidate is analyzed to extract its identification code (an image, a pattern, a binary code, etc.), which is matched with a list of known codes to remove falsely identified non-marker objects. Finally, these identified markers are processed to obtain their precise location in 3D world, usually by processing their shape.

Barcodes (Figure 1.2a) and QR codes (Figure 1.2b) are examples of well known markers. These markers are designed to contain a lot of information, but the detection is expected to be done in good conditions, such as in situations when the markers are facing the camera and are close to the viewer and not moving to avoid motion blur. If these conditions can be achieved, they can be used in AR applications [4], however, this is not possible in most AR applications, so more robust solutions are used.

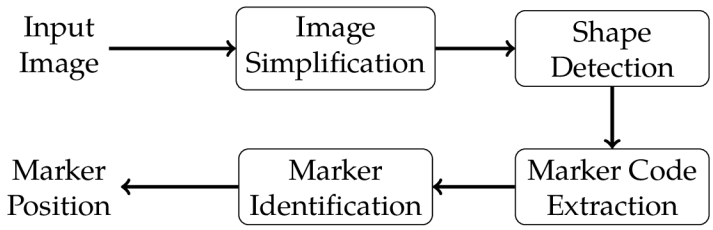


Figure 1.1: General workflow of algorithms detecting square markers

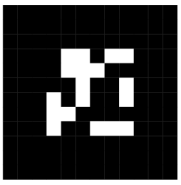
Square markers are very common in marker-based AR, since polygonal shapes are easy to detect in a thresholded image. Some of these solutions are ARToolKit [5], ARToolKitPlus [6], ARUco [7], ARUco3 [8], AprilTag [9], AprilTag2 [10], or ARTag [11]. These solutions take a grayscale image, and perform thresholding by comparing the intensity of pixels with a fixed value (global thresholding [5, 8]) or use a different value for each pixel that depend on pixel's surrounding (adaptive thresholding [7, 10]). Global thresholding methods run faster, however, they are more prone to worse lighting conditions [8, 9]. Inner part of markers is used to distinguish between individual markers and to filter out false positives. Arbitrary images [5] (see Figure 1.2c) are used as well as binary codes [11, 7] (see Figures 1.2d, 1.2e, and 1.2f), the former are more friendly for the users, while the later are more robust. This robustness is achieved by detecting errors in registered code and correcting them using Hamming codes [10] or specialized dictionaries [7].

More complicated shapes increase the stability of tracking if the conditions are not perfect, for example, circular and elliptic markers (see Figures 1.2g and 1.2h) are more robust to occlusions, since only a small part of their contour is necessary to compute its relative pose to the viewer [12, 13]. Shape of markers may also be irregular, unimportant, or completely missing. RUNE Tag [14] (see Figure 1.2i) is a marker that is composed of dots arranged in concentric circles, and since these dots are not connected, there is no contour to detect. ReacTIVision markers [15] (see Figure 1.2j) are composed of irregularly placed black dots on white background and white dots on background. All of these solutions are designed to be detect even if some parts of them are occluded, and use a variant of a binary pattern to distinguish between

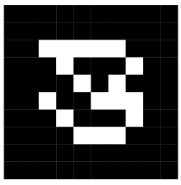
(a) EAN 13 Barcode, courtesy of Wikipedia.org

(b) QR Code

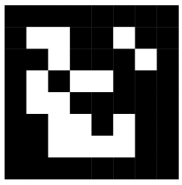
(c) ARToolKit [5]



(d) ARToolKitPlus [6]



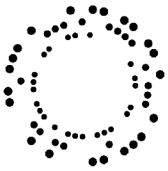
(e) ARTag [11]



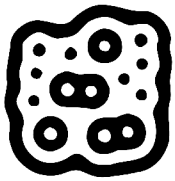
(f) ARUco [7] ARUco3 [8] AprilTag [9] AprilTag2 [10]

(g) Intersence [12]

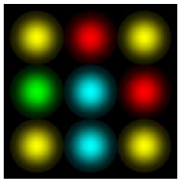
(h) Elipses [13]



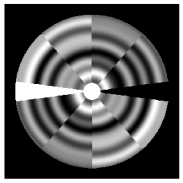
(i) RUNE Tag [14]



(j) reacTIVision [15]



(k) Mono-spectrum marker [16]



(l) Fourier Tag [17]

(m) CC Tag [18]

Figure 1.2: Various markers for detection and registration of objects

markers. However, their processing time is higher when compared to the time required to detect and identify square markers.

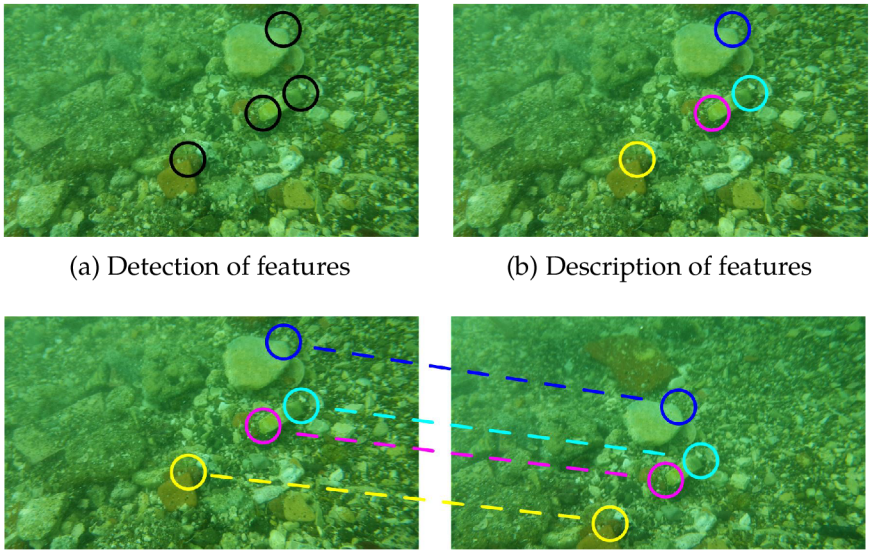
Much less work was done in the field of designing and detecting markers in blurry or foggy visual conditions. Mono-spectrum markers [16] (see Figure 1.2k) are composed of dots that are blurred, and thus they do not contain high-frequency information. Since the low-frequency information is preserved in blurred images, they are more robust to bad focus and camera motion. Similar property is exploited in Fourier tags [17] (see Figure 1.2l) that encode marker identification in frequency domain. CC Tags [18] (see Figure 1.2m) are circular markers that are robust to both occlusions and motion.

1.2 Marker-less detection of objects

Methods that detect objects using markers require the area to be populated with these markers before actual tracking is performed. Although this is not a problem in controlled environments like laboratories or museums, it is impractical or difficult to do so in large outdoor environments or in areas that are hard to access, e.g., under water. In such situations, marker-less tracking techniques are a viable option.

Techniques that recognize objects solely according to their appearance are prone to bad lighting conditions, so they are suitable only for controlled environments, similarly as techniques that detect elementary features (e.g., edges), since they work well only with certain classes of objects [19]. More general solutions usually contain the following three steps that are required for successful object registration. Input image is first searched for important features like corners or edges, which are in the second step labeled with a vector that describes appearance of each feature, and these vectors are matched in the third step to recognize the same features (and thus also objects) in different images, as is illustrated in Figure 1.3. All of these steps are required to produce equal results in every condition, and be robust to changes in illumination, and scale and rotation of images.

One of the first algorithm for natural feature detection used in AR applications was SIFT [19, 20], which uses difference of Gaussians to detect important features in the image, and describes each of them with a vector of 128 floating point numbers computed from magnitude and orientation of the gradient around it. SURF detector [21] computes approximation of Hessian to detect important features, and builds its



(a) Detection of features

(b) Description of features

(c) Matching corresponding features

Figure 1.3: Three steps in marker-less tracking: detection, description, and matching.

descriptor on a response of Haar wavelets. Both of these solutions use floating point arithmetic, which is unfortunately very computationally demanding.

There are also solutions that use mainly integer operations and binary comparisons to detect and register features. FAST [22] is an algorithm for detection of corners in images that uses machine learning approach to construct a decision tree with the fastest order of comparisons around each pixel, and since it only compares intensities, it is very quick. This algorithm was improved in FASTER [23], which further mutates the decision tree by adding more options, and in AGAST [24], which switches between multiple optimized decision trees for each pixel. FAST algorithm itself does not describe the features, but it can be used in combination with separate feature descriptors. BRIEF [25] is a descriptor that assigns each feature a vector of 128 bits representing a binary result of 128 comparisons of pixels around a feature. Since it is a binary vector, matching features is faster than in case of SIFT and SURF, however, the implementation presented in [25] is not invariant

to rotations and scales. BRISK [26] is a natural feature detector that uses AGAST to detect features, and compares pixels in circular patters around features to achieve rotation and scale invariance. FREAK [27] is a detector similar to BRISK, but its circular pattern is based on human retina. ORB detector [28] is composed of FAST detector and BRIEF descriptor that are optimized to provide robust detection invariant to rotations and scales.

Many authors compare existing algorithms for detection and registration of features that work in good conditions on data from underwater areas, for the purpose of general navigation under water [29] or specifically for navigating underwater vehicles [30, 31, 32]. However, there is a very little work on detection, description, or matching of features that is designed for underwater environments. LBP-SURF [33] adapts feature descriptor of SURF algorithm to be more robust to changes in colors under water, and compares it with standard SIFT and SURF. The problem of matching features between land images and underwater images caused by different imaging conditions is tackled in a work of Oliver et al. [34, 35].

1.3 Improving images taken in reduced visibility conditions

Images taken under water are often degraded similarly as images taken on land. Problems like noise or blurring are solved with well known filters [36, 37, 38], but other issues like reduced visibility caused by turbidity or change in colors due to uneven absorption of light require more sophisticated techniques, similar to those that are used for improving foggy and hazy images taken on land. Some of these techniques utilize additional data about the scene, e.g., Treibitz et al. [39] discusses advantages and disadvantages of using polarization filters to obtain several images of the same scene with different polarization of the light to suppress the effect of fog in final image, and Kopf et al. [40] uses depth information at each pixel to relight the scene and remove the effect of haze. However, for the purpose of mobile underwater AR applications, such solutions are not an option, because they require a specialized hardware.

(a) Original image (b) Fattal [43] (c) He et al. [41] (d) Zhu et al. [42]

Figure 1.4: Comparison of dehazing algorithms

The problem of removing haze from individual images became a more important topic of research in the last decade. He et al. [41] discovers that most regions of haze-free images have at least one color channel (red, green, or blue) that contains very low values. They name this channel a dark channel, and propose a method for dehazing called dark channel prior, which is based on this information. Zhu et al. [42] describes a similar method that is based on a difference between saturation and value channels of pixels represented in HSV color space. In 2008, Fattal [43] works with an assumption that illumination of objects in a scene and depth of these objects are two statistically uncorrelated functions, and uses methods for independent component analysis to separate these functions and to recover image without haze. In 2014, Fattal [44] describes another technique that is based on color lines [45], which processes small patches in input images with colors that belong to the same color line, and uses properties of this color line to estimate the amount of haze in that patch. For results of these works, see Figure 1.4.

In addition to these general solutions, some methods are more suitable for the mobile AR applications. Zhang et al. [46] not only describes a method that tests several possible solutions and chooses the best solution for each pixel of input image individually, but most importantly, he provides an implementation for GPUs that runs in real time. Li et al. [47] builds on methods that compute a depth of an image from a video sequence, and develops a method that simultaneously reconstructs the scene and removes fog in the scene, using the depth cues from scene reconstruction for removing fog and vice versa. Ancuti et al. [48] focuses specifically at a detection of SIFT features for

matching two images, and proposes a method that enhances contrast in images to recover and match as many features as possible.

In recent years, a lot of attention has been focused on neural networks. Cai et al. [49] is inspired by techniques based on dark channel, enhancing contrast, attenuation of colors, and hue disparity, and designs a convolutional neural network (CNN) that obtains a transmission map required for removing fog. Ren et al. [50] designs two CNNs, a first that obtains an approximation of a coarse transmission map, and a second that uses this information as an input to compute the precise transmission map. These CNNs do not depend on any assumptions or prior knowledge of input images, however, they need a substantial amount of training images with known ground truth to train them. This problem is tackled by Li et al. [51], who proposes a neural network for improving underwater images without needing a set of training images with a known ground truth.

1.3.1 Improving underwater images

Restoration of underwater images represents a greater challenge than restoration of images taken on land, due to the fact that underwater images are degraded by much more phenomena than other images, most importantly by turbidity, floating particles, and uneven absorption of different color channels of light. Ancuti et al. [52] proposes a system that derives multiple images from the input image using common image improving techniques like white balancing and contrast enhancement, and fuses them together. Li et al. [53] creates a pipeline of several filters to improve images degraded by uneven illumination, noise, turbidity, and uneven absorption of colors.

Dark channel prior provides very good results in images taken outside of water, but it is difficult to use it for images taken in large depths under water due to the missing red and green channels, although Chiang et al. [54] successfully applies it to estimate the depth of objects in the input image, and reduce the effect of turbidity and uneven illumination. Other solutions adapt dark channel prior to work with underwater images. Carlevaris-Bianco et al. [55] uses the difference between the maxima in red, green and blue channels to get a coarse estimate of the depth in the image, which is refined by natural image matting algorithms as in the work by He et al. [41], Drews et al.

(a) Original image (b) He et al. [41] (c) Ancuti et al. [52] (d) Gao et al. [57]

Figure 1.5: Results of dehazing algorithms on underwater images

[56] suggests ignoring the red channel completely and use only green and blue channels to estimate the depth in the image, and Gao et al. [57] presents bright channel prior, which operates with the original red channel and inverted green and blue channels (see Figure 1.5 for comparison of this algorithm with He et al. [41] and Ancuti et al. [52]).

These methods try to estimate the depth from the input image, but the depth can be obtained also using other methods. Drews et al. [58] focuses on processing a video instead of using individual images, reconstructs the scene to obtain the depth, and uses it again to reduce the effects of turbidity, Babaee et al. [59] proposes using a sonar in environments with high turbidity to obtain 3D positions of objects, and Cho et al. [60] uses a depth that is precisely measured in a few points in the input image, and uses incremental Gaussian process to estimate the depth in the rest of the image.

The problem of improving images taken in bad visibility conditions for the purpose of improving computer vision results was examined in several scenarios. Agarwal et al. [61] compares methods for increasing contrast of images taken in smoke occluded environments to improve visual odometry based on a sequence of RGB-D images. Cesar et al. [62] focuses on marker-based tracking, and evaluates the performance of detecting algorithms in laboratory in artificially simulated bad underwater conditions. Other authors work on registration of images, and build their comparisons on a number of detected and matched SIFT features. Andono et al. [63] compares contrast enhancing techniques to match a sequence of underwater images, Ancuti et al. [64] presents an algorithm for removing haze in foggy images to match images taken in different views, and Gao et al. [57] compares his method with other methods using a number of detected Canny edges in addition to the number of detected SIFT features. Image improving methods were

also used to improve the results of reconstructed underwater objects using photogrammetry in the work of Agrafiotis et al. [65] or Marino et al. [66].

1.4 Tracking and SLAM

Tracking location of a device in an unknown 3D environment is very close to problem of Simultaneous Localization and Mapping (SLAM). This is a problem of tracking a pose of an object (robot, mobile device, etc.) in an unknown area, while simultaneously creating a map of this area. SLAM techniques are composed of solutions to the following subproblems: initialization of the system (initial map and initial pose), identification and recognition of other objects (optical flow, matching important features), tracking (computing a pose from multiple views), relocalization (recognition of visited places), mapping (creating a 3D map of features), global map optimization (simplification and compression of the map), pose-graph optimization (simplification and compression of object's path), and loop closing (connection of the same objects in the map and in the pose graph with subsequent optimization when a loop in object's path is detected).

There is a lot of solutions for SLAM in good visual conditions on land, so this section focuses only on those that are targeted to mobile devices. Survey of Taketomi et al. [67] discusses general concepts of existing solutions of mobile SLAM techniques, while survey of Younes et al. [68] focuses more deeply on 23 chosen methods. Klein et al. [69] describes a concept of Parallel Tracking and Mapping (PTAM) that proposes splitting of tracking and mapping solutions into two separate threads to solve these problems at different rates, usually every frame for tracking and every few seconds for mapping. Since the introduction of PTAM, this concept appears in most of following solutions. ORB-SLAM [70] represents a feature-based solution to SLAM that detects ORB features [28] in images, which are used for tracking and creating the map, and thanks to global map optimization and loop closing techniques, this method is able to run in real time in large scale environments. Unlike feature-based methods, direct methods work with images without detecting important features. Schöps et al. [71] presents a direct method for SLAM, and is able to create a map of a

small room in real time using a mobile device. LSD-SLAM [72] extends this technique with map optimization algorithms to run in real-time in outdoor environments, similarly as ORB-SLAM. With introduction of cheap RGB-D sensors (e.g., Microsoft Kinect), the map of environment can be constructed using data from depth sensor, as is mentioned by Kahler et al. [73] that is able to scan areas in real time using GPU acceleration. Inertial sensors (accelerometers and gyroscopes) can be used in cooperation with visual data from camera to improve tracking, for example, Piao et al. [74] incorporates inertial sensors into ORB-SLAM, and Li et al. [75] similarly extends KLT-SLAM that uses KLT features [76, 77] instead of ORB features. SLAM can also be performed using artificial markers to simplify detection and registration of important features, like in a work of Muñoz-Salinas et al. [78] that describes an algorithm that uses ARUco marker detector [7].

Underwater SLAM is different though, since most of the techniques focus on remotely operated vehicles (ROVs) and autonomous underwater vehicles (AUVs). Such devices are provided with advanced sensors for underwater localization, such as sonars or Doppler velocity loggers (DVLs), which are not suitable for use by divers that carry mobile devices. Guth et al. [79] presents a survey of 20 techniques for underwater SLAM, all of them for ROVs and AUVs, and compares their performance, accuracy, and simplicity of implementation, Hidalgo et al. [80] presents a similar survey, but focuses more on the whole families of underwater SLAM solutions than on individual solutions, and Ferreira et al. [81] reviews solutions that fuse visual and acoustic sensors. Ferreira et al. [82] in another work presents a solution that uses only visual data to map seabed by tracking and matching SURF features. Zhang et al. [83] evaluates ORB-SLAM in underwater videos, and compares the results with videos that are preprocessed by image improving algorithms CLAHE, Fusion [52], and dark channel prior [41].

1.5 Underwater augmented reality

Underwater augmented reality is mostly unexplored area with a very few solutions for controlled environments like swimming pools, and even fewer for an open sea. One of the first underwater AR solutions

was proposed was by Gallagher [84] for Navy divers. From more resent solutions, Morales et al. [85] presents an experimental AR system that provides visual aids to increase commercial divers' capability to detect, perceive, and understand elements in underwater environments, and Brown et al. [86] presents an underwater AR system to help divers with navigation and identification of fish species. Regarding solutions for swimming pools, AREEF system allows people to play a simple game under water using a see-through display connected to a mobile PC. This system was first presented by Blum et al. [87], and then extended by Oppermann et al. [88] to allow interaction between more than one person, and again by Oppermann et al. [89] to educate children using an underwater serious game with waterproof tablets. A similar system is realized also with an autonomous device called DOPLHYN [90].

2 Aims of the thesis

2.1 Contributions and expected publications

As already mentioned in Introduction, my research is focused on detection of objects and tracking location of mobile devices under water to create a system for presenting European underwater cultural heritage using AR that is robust to visual conditions in sea. Expected contributions are as follows:

- **Detection of markers in bad visibility conditions under water**: The first objective of my research is focused on detecting objects using square markers for real-time AR applications running in bad underwater visibility conditions on mobile devices. Methods for such detection already exist and work well in good visibility conditions on land, however, solutions working in sea are still unexplored, since current research has been performed only in simplified laboratory conditions or with off-line methods. The contribution of my research in this field will be:

  - evaluation of performance of current real-time image improving algorithms and marker detecting algorithms in real underwater conditions of open sea;
  - design of new image improving algorithms and marker detecting algorithms that perform well in sea and are suitable for AR.

  The main novelty of this part of my research lies in developing algorithms for detecting markers that are robust to underwater conditions, and are suitable for AR applications that run on mobile devices. Current solutions are not targeted for real-time AR, are computationally expensive, or are designed only for clean areas.

- **Hybrid visual-inertial tracking of location of mobile devices in harsh underwater conditions**: The second objective of my thesis is on tracking position of mobile devices, which is required for AR applications. Current solutions already work with

data from visual and inertial sensors (as well as other data if available, like GPS or WiFi), but they are again designed to work in clean environments on land, or are for large underwater vehicles. No research was done to identify complications that are connected with robust tracking of mobile devices for AR applications running under water and to find solutions to them. My contribution in this field will be:

- comparison of current approaches for visual-inertial navigation using real underwater data and artificial data simulating various levels of bad visibility in laboratory conditions;
- new solution that combine the data from marker-based tracking, marker-less tracking, and inertial sensors to create a localization system for a mobile underwater AR adapted for visibility conditions in open sea;

Results of this part of research will provide new solutions for tracking diver's pose required for AR applications that will be robust to conditions under water. Similarly to the first objective, current solutions are not aimed at real-time underwater AR.

- **Underwater AR system for application in archaeology and in presenting cultural heritage**: The third objective is to develop a system that exploits benefits of AR to help underwater archaeologists with orientation on site, and present history of important underwater places to diving tourists. Apart of solutions for detecting objects and tracking divers, which are the subject of the previous two objectives, my contribution will also be in the field of designing techniques for interaction between the system and divers under water, and in the field of presenting information about underwater sites. Similar systems already exist, but are designed for use on land or in clean water of swimming pools.

The following results are expected to be published or are already published:

- two publications about improving underwater images for marker-based tracking (one already presented at *GCH 2018*, and one already published in *Frontiers in Robotics and AI*);

- one publication about a new marker detecting algorithm designed for underwater environments (submitted to *Remote Sensing*);
- one publication about a new solution for tracking device location in bad underwater visibility and a its comparison with other methods in simulated laboratory conditions (to be sent to *ISMAR 2019* or a similar conference);
- one publication about a new solution for visual-inertial localization under water tested in real underwater scenarios (to be sent as an extension of ISMAR paper to *TVCG*, to *Remote Sensing* journal, or a similar journal);
- one publication about the architecture of the whole underwater system and its evaluation (to be sent to a special issue of Computer Graphics and Applications called *Graphics and Virtual Environments for Serious Games*).

2.2 Study plan

I expect to finish this research in two years according to the following plan:

- **Spring 2019**
  - Design of a visual-inertial system for tracking location of a mobile device in underwater conditions, its evaluation in laboratory conditions, and submitting the results to ISMAR 2019;
  - Implementation of this system as an AR application for preliminary tests done under water in Baiae in June 2019, and acquisition of new testing data.
- **Autumn 2019**
  - Design of the final version of the system, its evaluation in Baiae in Oct 2019, and acquisition of new data;
  - Submitting a paper about the whole system to the special issue of *Computer Graphics and Applications*;
  - Submitting a paper to *TVCG*, *Remote Sensing*, or a similar journal about the improvements of visual-inertial system and results of tests performed in Baiae in June 2019 and October 2019.
- **Spring 2020**
  - Thesis writing and submission
- **Autumn 2020**
  - Thesis defence

3 Achieved results

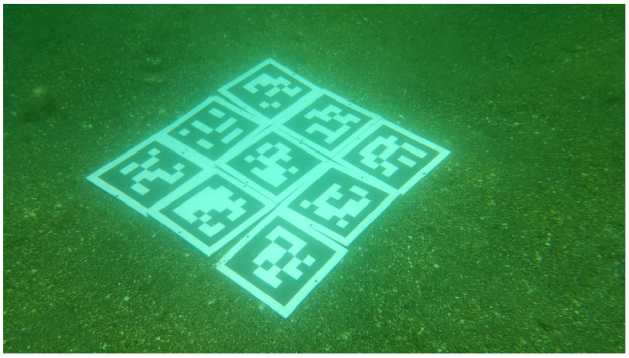
I already achieved goals of the first objective of my expected contributions, and initial results of the second and the third objective. Namely, I succeeded in improving underwater images for detection of markers, designing a marker detecting algorithm that is robust to underwater conditions, evaluation of a simple solution for visual-inertial tracking, and development of an initial version of underwater localization and tracking system. Although data used in my experiment were obtained in cooperation with colleagues of iMareCulture project, the implementation and analysis is my own work.

3.1 Improving underwater images for detection of markers

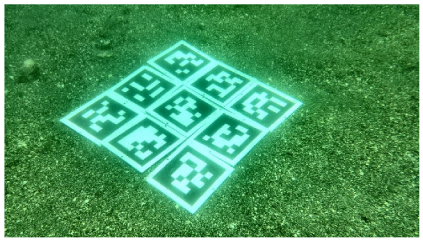
The first achieved result is in the field of improving underwater images. For initial evaluation of marker detecting and image improving algorithms, divers recorded videos with markers at several places of Mediterranean sea in different visibility conditions. These videos were processed by existing image improving algorithms, and passed to different marker detectors.

The following real-time image improving algorithms were tested: Gaussian filter, median filter, bilateral filter, histogram equalization, CLAHE, white balancing, and debluring; as well as the following offline algorithms: Fusion [52], BCP [57], ACE [91], and SP [92]. Regarding marker detecting algorithms, we compared the following algorithms: ARToolKit [5], ARUco [7], and AprilTag2 [10]. However, preliminary results showed that Gaussian filter, median filter, bilateral filter, histogram equalization filter, and ARToolKit detector perform very poorly, so they were discarded from further extensive evaluation.

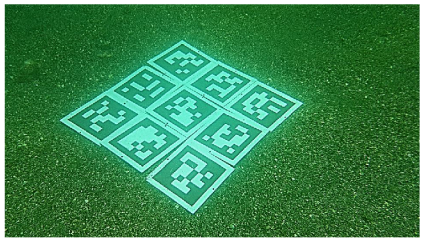
Results showed that image improving algorithms can improve detection of markers performed by ARUco detector. Detection of AprilTag2 was not improved, which hints that this algorithm is well suited for bad visibility conditions, however, its processing time was very high, and it could not run in real time on a mobile device. The results of offline algorithms were approximately the same as the results of the



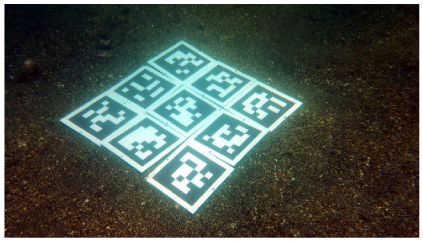
(a) Original image



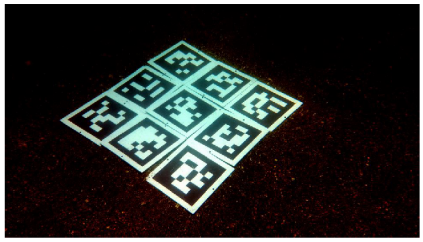
(b) CLAHE



(c) Debluring



(d) White balancing



(e) MBUWWB

Figure 3.1: Results of real-time image improving techniques when applied to underwater images

best real-time algorithms, which showed that simple image improving algorithms are sufficient.

White balancing algorithm performed very well, both in terms of number of detected markers and processing speed, so it was improved

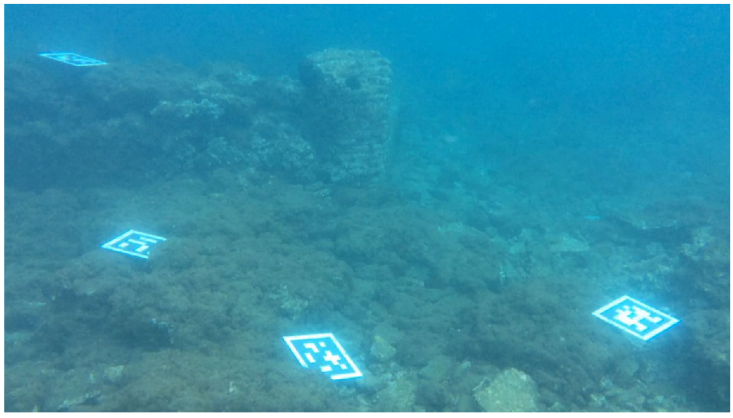
to cooperate with the algorithm for detection of markers to adapt better to underwater conditions. The improvement was based on the fact that white balancing algorithms need to find black and white colors in the image, and since the markers are black and white, the markers detected in the previous frame are used as reference colors for white balancing. This method, which we called Marker-Based UnderWater White Balancing (MBUWWB), see Figure 3.1, performed better than other image improving algorithms.

My results confirmed that properly designed methods for improving images taken in bad visibility conditions can help existing computer vision algorithms with detecting objects. The results were presented in *Impact of Dehazing on Underwater Marker Detection for Augmented Reality* and in *Improving Marker-Based Tracking for Augmented Reality in Underwater Environments*. Additionally, a simple AR application that detected markers and improved images in real time was tested in Baiae in June 2018. The divers claimed that the markers were detected from further distances when image improving techniques were applied, which was then verified and confirmed offline from recorded data.

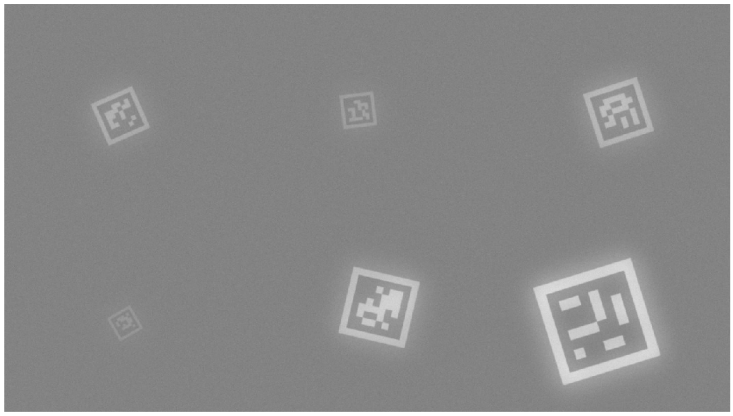
3.2 Detection of markers under water

Based on the previous results, ARUco marker detector was examined to find which parts are affected by image improving algorithms. It was found that the parts influenced by underwater conditions are image thresholding, detection and filtering of contours, and marker identification. These parts were adapted were incorporated into a new version of this detector called Underwater ARUco (UWARUco), which changes the methods for thresholding and marker identification, and speeds up the detection by a fast identification and exclusion of parts in input image that do not contain any marker.

In addition to this, a new algorithm was developed to generate synthetic images with markers in simulated underwater environments, see Figure 3.2. Unlike with images taken in real environments, it is easy to obtain a ground truth result of detection of markers in generated images. Thanks to it, it was found that the most problems with underwater detection are caused by high noise in input image and unequal



(a) Real underwater images with markers



(b) Generated underwater images with markers

Figure 3.2: Real and generated underwater images with markers. Notice a small glow around markers in both images

visibility conditions at each marker when markers are at different depths, which helped with the development of UWARUco.

My research helped to identify problematic parts of existing solutions, and proved that solutions adapted to underwater environments perform better than solutions that only improves underwater images, which still work better than solutions that are not adapted to underwater conditions at all. These results are presented in publication

Detecting Square Markers in Underwater Environments, their evaluation in a real AR application is expected to be done in Baiae in June 2019.

3.3 Underwater localization and tracking system

The system for underwater tracking and localization is still under construction, and currently it contains three components: camera calibration, camera recorder, and a simple AR demo. Other components are expected to be implemented during 2019 to be tested in Baiae in June and in October.

Camera calibration app performs a basic camera calibration using functions of OpenCV. This calibration needs to be done under water, since the optic properties of water are different from air, but fortunately, it is easy to recalculate the calibration performed in one environment to work in another environment, as presented in [93] and experimentally verified from recorded data.

Camera recorder app records data to be processed offline. Currently, the application saves a video and data from accelerometer, gyroscope, and magnetometer, and stores them without any compression together with their time stamps, so that they could be synchronized when they are used for testing. The application is optimized to record as much data as possible by saving the data stream concurrently into both internal memory and external SD card. In June 2018 in Baiae, data recorded by this application helped to understand effects of glowing markers and sensor noise, which were subsequently used to generate synthetic images.

AR demo app is an application that detects markers and renders 3D objects into AR view. It is also able to run real-time image improving techniques and use gyroscope to keep location of the device when the markers are lost. Apart of results already mentioned in Section 3.1, preliminary tests performed in July 2018 in Epidauros showed that even a simple use of gyroscope can greatly improve user experience under water, unlike straightforward integration of data from accelerometer which resulted in a huge error in position, as expected. Proper evaluation of gyroscope and other sensors is planned in 2019.

Applications were tested with multiple versions of user interfaces. It was found that waterproof housings, which allow mobile devices

Figure 3.3: Interface of AR demo app simplified for manipulation under water, its GUI was reduced to three large buttons placed in the corners of the screen. The background shows a model of ancient Villa a Protiro, whose remains currently lie several meters under water. Recorded during tests in Baiae in June 2018.

to work under water, limit the number of options that divers have for direct interaction with devices. Interfaces were simplified to contain only a few large buttons, and were rid of gestures. An example of such interface is in Figure 3.3.

3.4 Publications

The results of my research were presented in the following publications:

- Fabio Bruno, Antonio Lagudi, Gerardo Ritacco, Panagiotis Agrafiotis, Dimitrios Skarlatos, **Jan Čejka**, Pavel Kouřil, Fotis Liarokapis, Oliver Philpin-Briscoe, Charalambos Poullis, Sudhir Mudur, Bart Simon: *Development and integration of digital technologies addressed to raise awareness and access to European underwater cultural heritage. An overview of the H2020 i-MARECULTURE project*, Proceedings of OCEANS 2017, Aberdeen, 2017.

  Publication about the whole iMareCulture project.

My participation is a section about an initial version of augmented reality application, 10 %.

- Marek Žuži, **Jan Čejka**, Fabio Bruno, Dimitrios Skarlatos, Fotis Liarokapis: *Impact of Dehazing on Underwater Marker Detection for Augmented Reality*, Frontiers in Robotics and AI, 2018.

  Publication about improving images for detecting markers, focusing on offline methods.

  My participation is on supervision (it is based on Marek Žuži's Master thesis that I supervised), development of a tool for evaluation, and design of tests, 40 %.

- **Jan Čejka**, Marek Žuži, Panagiotis Agrafiotis, Dimitrios Skarlatos, Fabio Bruno, Fotis Liarokapis: *Improving Marker-Based Tracking for Augmented Reality in Underwater Environments*, Proceedings of Eurographics Workshop on Graphics and Cultural Heritage, 2018.

  Publication is also about image improving algorithms for detecting markers, but this is focused on real-time methods. It also presents my new method for improving images – Marker-Based Underwater White Balancing.

  My participation is on design of the algorithm, design of a method for assessing visual conditions, comparison with other algorithms, and writing the paper, 85 %.

- **Jan Čejka**, Fabio Bruno, Dimitrios Skarlatos, Fotis Liarokapis: *Detecting Square Markers in Underwater Environments*, submitted to Remote Sensing journal.

  Publication is about new method for detecting markers under water (UWARUco), and a method for generating synthetic underwater images that contain markers.

  My participation is on design of the marker detector, design of the method for generating images with markers, comparison with other algorithms, and writing the paper, 85 %.

Appendix

The following papers are attached to this proposal:

- Bruno et al.: *Development and integration of digital technologies addressed to raise awareness and access to European underwater cultural heritage. An overview of the H2020 i-MARECULTURE project*
- Žuži et al.: *Impact of Dehazing on Underwater Marker Detection for Augmented Reality*
- Čejka et al.: *Improving Marker-Based Tracking for Augmented Reality in Underwater Environments*
- Čejka et al.: *Detecting Square Markers in Underwater Environments*

See Section 3.4 for more details.

Bibliography

1. EDNEY, Joanne; SPENNEMANN, Dirk H. R. Can Artificial Reef Wrecks Reduce Diver Impacts on Shipwrecks? The Management Dimension. *Journal of Maritime Archaeology*. 2015, vol. 10, pp. 141–157. ISSN 1557-2293. Available from DOI: `10.1007/s11457-015-9140-5`.
2. VLAHAKIS, Vassilios; IOANNIDIS, Nikolaos; KARIGIANNIS, John; TSOTROS, Manolis; GOUNARIS, Michael; STRICKER, Didier; GLEUE, Tim; DAEHNE, Patrick; ALMEIDA, Luís. Archeoguide: An Augmented Reality Guide for Archaeological Sites. *IEEE Computer Graphics and Applications*. 2002, vol. 22, pp. 52–60. ISSN 0272-1716. Available from DOI: `10.1109/MCG.2002.1028726`.
3. PANOU, Chris; RAGIA, Lemonia; DIMELLI, Despoina; MANIA, Katerina. An Architecture for Mobile Outdoors Augmented Reality for Cultural Heritage. *ISPRS International Journal of Geo-Information*. 2018, vol. 7, no. 12. ISSN 2220-9964. Available from DOI: `10.3390/ijgi7120463`.
4. KAN, Tai-Wei; TENG, Chin-Hung; CHOU, Wen-Shou. Applying QR Code in Augmented Reality Applications. In: *Proceedings of the 8th International Conference on Virtual Reality Continuum and Its Applications in Industry*. Yokohama, Japan: ACM, 2009, pp. 253–257. ISBN 978-1-60558-912-1. Available from DOI: `10.1145/1670252.1670305`.
5. KATO, Hirokazu; BILLINGHURST, Mark. Marker Tracking and HMD Calibration for a Video-Based Augmented Reality Conferencing System. In: *Augmented Reality, 1999. (IWAR '99) Proceedings. 2nd IEEE and ACM International Workshop on*. 1999, pp. 85–94. Available from DOI: `10.1109/IWAR.1999.803809`.
6. WAGNER, Daniel; SCHMALSTIEG, Dieter. ARToolKitPlus for Pose Tracking on Mobile Devices. In: *Proceedings of 12th Computer Vision Winter Workshop CVWW07*. 2007.
7. GARRIDO-JURADO, Sergio; MUÑOZ-SALINAS, Rafael; MADRID-CUEVAS, Francisco José; MARÍN-JIMÉNEZ, Manuel Jesús. Automatic Generation and Detection of Highly Reliable Fiducial Markers Under Occlusion. *Pattern Recognition*. 2014, vol. 47, pp. 2280–2292.

Available from DOI: http://dx.doi.org/10.1016/j.patcog.2014.01.005.

8. ROMERO-RAMIREZ, Francisco J.; MUÑOZ-SALINAS, Rafael; MEDINA-CARNICER, Rafael. Speeded up detection of squared fiducial markers. *Image and Vision Computing*. 2018, vol. 76, pp. 38–47. ISSN 0262-8856. Available from DOI: https://doi.org/10.1016/j.imavis.2018.05.004.

9. OLSON, Edwin. AprilTag: A robust and flexible visual fiducial system. In: *2011 IEEE International Conference on Robotics and Automation*. 2011, pp. 3400–3407. ISSN 1050-4729. Available from DOI: 10.1109/ICRA.2011.5979561.

10. WANG, John; OLSON, Edwin. AprilTag 2: Efficient and robust fiducial detection. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2016.

11. FIALA, Mark. ARTag, a Fiducial Marker System Using Digital Techniques. In: *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 590–596. Available from DOI: 10.1109/CVPR.2005.74.

12. NAIMARK, Leonid; FOXLIN, Eric. Circular Data Matrix Fiducial System and Robust Image Processing for a Wearable Vision-Inertial Self-Tracker. In: *Proceedings of International Symposium on Mixed and Augmented Reality*. 2002, pp. 27–36. Available from DOI: 10.1109/ISMAR.2002.1115065.

13. KÖHLER, Johannes; PAGANI, Alain; STRICKER, Didier. Robust Detection and Identification of Partially Occluded Circular Markers. In: *VISAPP 2010 - Proceedings of the Fifth International Conference on Computer Vision Theory and Applications*. 2010, pp. 387–392.

14. BERGAMASCO, Filippo; ALBARELLI, Andrea; COSMO, Luca; RODOLÁ, Emanuele; TORSELLO, Andrea. An Accurate and Robust Artificial Marker Based on Cyclic Codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2016, vol. 38, pp. 2359–2373. Available from DOI: 10.1109/TPAMI.2016.2519024.

15. BENCINA, Ross; KALTENBRUNNER, Martin; JORDA, Sergi. Improved Topological Fiducial Tracking in the reacTIVision System. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*. 2005. Available from DOI: 10.1109/CVPR.2005.475.

16. TOYOURA, Masahiro; ARUGA, Haruhito; TURK, Matthew; MAO, Xiaoyang. Detecting Markers in Blurred and Defocused Images. In: *2013 International Conference on Cyberworlds*. 2013, pp. 183–190. Available from DOI: 10.1109/CW.2013.58.

17. XU, Anqi; DUDEK, Gregory. Fourier Tag: A Smoothly Degradable Fiducial Marker System with Configurable Payload Capacity. In: *2011 Canadian Conference on Computer and Robot Vision*. 2011, pp. 40–47. Available from DOI: 10.1109/CRV.2011.13.

18. CALVET, Lilian; GURDJOS, Pierre; GRIWODZ, Carsten; GASPARINI, Simone. Detection and Accurate Localization of Circular Fiducials under Highly Challenging Conditions. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 562–570. ISSN 1063-6919. Available from DOI: 10.1109/CVPR.2016.67.

19. LOWE, David G. Object recognition from local scale-invariant features. In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. 1999, vol. 2, pp. 1150–1157. Available from DOI: 10.1109/ICCV.1999.790410.

20. GORDON, Iryna; LOWE, David G. Scene modelling, recognition and tracking with invariant image features. In: *Third IEEE and ACM International Symposium on Mixed and Augmented Reality*. 2004, pp. 110–119. Available from DOI: 10.1109/ISMAR.2004.53.

21. BAY, Herbert; TUYTELAARS, Tinne; VAN GOOL, Luc. SURF: Speeded Up Robust Features. In: LEONARDIS, Aleš; BISCHOF, Horst; PINZ, Axel (eds.). *Computer Vision – ECCV 2006*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 404–417. ISBN 978-3-540-33833-8.

22. ROSTEN, Edward; DRUMMOND, Tom. Machine Learning for High-Speed Corner Detection. In: LEONARDIS, Aleš; BISCHOF, Horst; PINZ, Axel (eds.). *Computer Vision – ECCV 2006*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 430–443. ISBN 978-3-540-33833-8.

23. ROSTEN, Edward; PORTER, Reid; DRUMMOND, Tom. Faster and Better: A Machine Learning Approach to Corner Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2010, vol. 32, pp. 105–119. ISSN 0162-8828. Available from DOI: 10.1109/TPAMI.2008.275.

24. MAIR, Elmar; HAGER, Gregory D.; BURSCHKA, Darius; SUPPA, Michael; HIRZINGER, Gerhard. Adaptive and Generic Corner Detection Based on the Accelerated Segment Test. In: DANIILIDIS, Kostas; MARAGOS, Petros; PARAGIOS, Nikos (eds.). *Computer Vision – ECCV 2010*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 183–196. ISBN 978-3-642-15552-9.

25. CALONDER, Michael; LEPETIT, Vincent; STRECHA, Christoph; FUA, Pascal. BRIEF: Binary Robust Independent Elementary Features. In: DANIILIDIS, Kostas; MARAGOS, Petros; PARAGIOS, Nikos (eds.). *Computer Vision – ECCV 2010*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 778–792. ISBN 978-3-642-15561-1.

26. LEUTENEGGER, Stefan; CHLI, Margarita; SIEGWART, Y. Brisk: Binary robust invariant scalable keypoints. In: *In Computer Vision (ICCV), 2011 IEEE International Conference on*. 2011, pp. 2548–2555.

27. ORTIZ, Raphael. FREAK: Fast Retina Keypoint. In: *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Washington, DC, USA: IEEE Computer Society, 2012, pp. 510–517. ISBN 978-1-4673-1226-4.

28. RUBLEE, Ethan; RABAUD, Vincent; KONOLIGE, Kurt; BRADSKI, Gary. ORB: An efficient alternative to SIFT or SURF. In: *2011 International Conference on Computer Vision*. 2011, pp. 2564–2571. ISSN 2380-7504. Available from DOI: 10.1109/ICCV.2011.6126544.

29. ŻAK, Bogdan; HOŻYŃ, Stanisław. Local image features matching for real-time seabed tracking applications. *Journal of Marine Engineering & Technology*. 2017, vol. 16, pp. 273–282. Available from DOI: 10.1080/20464177.2017.1386266.

30. FERREIRA, Fausto; VERUGGIO, Gianmarco; CACCIA, Massimo; BRUZZONE, Gabriele. A comparison between different feature-based methods for ROV vision-based speed estimation. *IFAC Proceedings Volumes*. 2012, vol. 45, no. 5, pp. 325–330. ISSN 1474-6670.

Available from DOI: https://doi.org/10.3182/20120410-3-PT-4028.00054. 3rd IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles.

31. SHKURTI, Florian; REKLEITIS, Ioannis; DUDEK, Gregory. Feature Tracking Evaluation for Pose Estimation in Underwater Environments. In: *2011 Canadian Conference on Computer and Robot Vision*. 2011, pp. 160–167. Available from DOI: 10.1109/CRV.2011.28.

32. FERREIRA, Fausto; VERUGGIO, Gianmarco; CACCIA, Massimo; BRUZZONE, Gabriele. A survey on real-time motion estimation techniques for underwater robots. *Journal of Real-Time Image Processing*. 2016, vol. 11, pp. 693–711. ISSN 1861-8219. Available from DOI: 10.1007/s11554-014-0416-z.

33. PRABHAKAR, C. J.; KUMAR, Praveen. LBP-SURF Descriptor with Color Invariant and Texture Based Features for Underwater Images. In: *Proceedings of the Eighth Indian Conference on Computer Vision, Graphics and Image Processing*. New York, NY, USA: ACM, 2012, pp. 1–8. ISBN 978-1-4503-1660-6. Available from DOI: 10.1145/2425333.2425356.

34. OLIVER, Kenton; HOU, Weilin; WANG, Song. Image feature detection and matching in underwater conditions. In: 2010, vol. 7678, pp. 1–12. Available from DOI: 10.1117/12.852339.

35. OLIVER, Kenton; HOU, Weilin; WANG, Song. Feature matching in underwater environments using sparse linear combinations. In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*. 2010, pp. 60–67. ISSN 2160-7508. Available from DOI: 10.1109/CVPRW.2010.5543905.

36. WEISS, Ben. Fast Median and Bilateral Filtering. *ACM Transactions on Graphics*. 2006, vol. 25, no. 3, pp. 519–526. Available from DOI: 10.1145/1141911.1141918.

37. TSIRIKOLIAS, Kostas D. Low Level Image Processing and Analysis Using Radius Filters. *Digital Signal Processing*. 2016, vol. 50, pp. 72–83. Available from DOI: https://doi.org/10.1016/j.dsp.2015.12.001.

38. KRASULA, Lukáš; CALLET, Patrick Le; FLIEGEL, Karel; KLÍMA, Miloš. Quality Assessment of Sharpened Images: Challenges, Methodology, and Objective Metrics. *IEEE Transactions on Image Processing*. 2017, vol. 26, pp. 1496–1508. Available from DOI: 10.1109/TIP.2017.2651374.

39. TREIBITZ, Tali; SCHECHNER, Yoav Y. Polarization: Beneficial for visibility enhancement? In: *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2009, pp. 525–532. Available from DOI: 10.1109/CVPRW.2009.5206551.

40. KOPF, Johannes; NEUBERT, Boris; CHEN, Billy; COHEN, Michael; COHEN-OR, Daniel; DEUSSEN, Oliver; UYTTENDAELE, Matt; LISCHINSKI, Dani. Deep Photo: Model-based Photograph Enhancement and Viewing. *ACM Transactions on Graphics*. 2008, vol. 27, no. 5. Available from DOI: 10.1145/1409060.1409069.

41. HE, Kaiming; SUN, Jian; TANG, Xiaoou. Single Image Haze Removal Using Dark Channel Prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2011, vol. 33, no. 12, pp. 2341–2353. Available from DOI: 10.1109/TPAMI.2010.168.

42. ZHU, Qingsong; MAI, Jiaming; SHAO, Ling. A Fast Single Image Haze Removal Algorithm Using Color Attenuation Prior. *IEEE Transactions on Image Processing*. 2015, vol. 24, pp. 3522–3533. Available from DOI: 10.1109/TIP.2015.2446191.

43. FATTAL, Raanan. Single Image Dehazing. *ACM Transactions on Graphics*. 2008, vol. 27, no. 3. Available from DOI: 10.1145/1360612.1360671.

44. FATTAL, Raanan. Dehazing Using Color-Lines. *ACM Transactions on Graphics*. 2014, vol. 34. Available from DOI: 10.1145/2651362.

45. OMER, Ido; WERMAN, Michael. Color Lines: Image Specific Color Representation. In: *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. 2004, vol. 2, pp. 946–953. Available from DOI: 10.1109/CVPR.2004.1315267.

46. ZHANG, Jun; HU, Shiqiang. A GPU-accelerated real-time single image de-hazing method using pixel-level optimal de-hazing criterion. *Journal of Real-Time Image Processing*. 2014, vol. 9, pp. 661–672. Available from DOI: 10.1007/s11554-012-0244-y.

47. LI, Zhuwen; TAN, Ping; TAN, Robby T.; ZOU, Danping; ZHOU, Steven Zhiying; CHEONG, Loong-Fah. Simultaneous Video Defogging and Stereo Reconstruction. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 4988–4997. Available from DOI: 10.1109/CVPR.2015.7299133.

48. ANCUTI, Cosmin; CODRUTA, Ancuti. Effective Contrast-Based Dehazing for Robust Image Matching. 2014, vol. 11, pp. 1871–1875.

49. CAI, Bolun; XU, Xiangmin; JIA, Kui; QING, Chunmei; TAO, Dacheng. DehazeNet: An End-to-End System for Single Image Haze Removal. *IEEE Transactions on Image Processing*. 2016, vol. 25, pp. 5187–5198. Available from DOI: 10.1109/TIP.2016.2598681.

50. REN, Wenqi; LIU, Si; ZHANG, Hua; PAN, Jinshan; CAO, Xiaochun; YANG, Ming-Hsuan. Single Image Dehazing via Multi-scale Convolutional Neural Networks. In: *ECCV 2016*. Springer International Publishing, 2016, pp. 154–169. Available from DOI: 10.1007/978-3-319-46475-6_10.

51. LI, Chongyi; GUO, Jichang; GUO, Chunle. Emerging From Water: Underwater Image Color Correction Based on Weakly Supervised Color Transfer. *IEEE Signal Processing Letters*. 2018, vol. 25, pp. 323–327. Available from DOI: 10.1109/LSP.2018.2792050.

52. ANCUTI, Cosmin; ANCUTI, Codruta O.; HABER, Tom; BEKAERT, Philippe. Enhancing Underwater Images and Videos by Fusion. In: *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Washington, DC, USA: IEEE Computer Society, 2012, pp. 81–88.

53. LI, Yujie; LU, Huimin; ZHANG, Lifeng; LI, Jianru; SERIKAWA, Seiichi. Real-Time Visualization System for Deep-Sea Surveying. 2014, pp. 1–10.

54. CHIANG, John Y.; CHEN, Ying-Ching. Underwater Image Enhancement by Wavelength Compensation and Dehazing. *IEEE Transactions on Image Processing*. 2012, vol. 21, pp. 1756–1769. Available from DOI: 10.1109/TIP.2011.2179666.

55. CARLEVARIS-BIANCO, Nicholas; MOHAN, Anush; EUSTICE, Ryan M. Initial Results in Underwater Single Image Dehazing. In: *OCEANS 2010*. 2010, pp. 1–8. Available from DOI: 10.1109/OCEANS.2010.5664428.

56. DREWS, Paulo L.J.; NASCIMENTO, Erickson R.; BOTELHO, Silvia S.C.; CAMPOS, Mario Fernando Montenegro. Underwater Depth Estimation and Image Restoration Based on Single Images. *IEEE Computer Graphics and Applications*. 2016, vol. 36, pp. 24–35. Available from DOI: 10.1109/MCG.2016.26.

57. GAO, Yakun; LI, Haibin; WEN, Shuhuan. Restoration and Enhancement of Underwater Images Based on Bright Channel Prior. 2016, vol. 2016, pp. 1–15.

58. DREWS, Paulo; NASCIMENTO, Erickson R.; CAMPOS, Mario F. M.; ELFES, Alberto. Automatic Restoration of Underwater Monocular Sequences of Images. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2015, pp. 1058–1064. Available from DOI: 10.1109/IROS.2015.7353501.

59. BABAEE, Mohammadreza; NEGAHDARIPOUR, Shahriar. Improved Range Estimation and Underwater Image Enhancement Under Turbidity by Opti-Acoustic Stereo Imaging. In: *OCEANS 2015*. 2015, pp. 1–7. Available from DOI: 10.1109/OCEANS-Genova.2015.7271611.

60. CHO, Younggun; SHIN, Young-Sik; KIM, Ayoung. Online Depth Estimation and Application to Underwater Image Dehazing. In: *OCEANS 2016 MTS/IEEE Monterey*. 2016, pp. 1–7. Available from DOI: 10.1109/OCEANS.2016.7761109.

61. AGARWAL, Aditya; MATURANA, Daniel; SCHERER, Sebastian. *Visual Odometry in Smoke Occluded Environments*. Pittsburgh, PA, 2014. Technical report. Robotics Institute, Carnegie Mellon University.

62. SANTOS CESAR, Diego Brito dos; GAUDIG, Christopher; FRITSCHE, Martin; REIS, Marco A. dos; KIRCHNER, Frank. An Evaluation of Artificial Fiducial Markers in Underwater Environments. In: *OCEANS 2015*. 2015, pp. 1–6. Available from DOI: 10.1109/OCEANS-Genova.2015.7271491.

63. ANDONO, Pulung Nurtantio; PURNAMA, I Ketut Eddy; HARIADI, Mochamad. Underwater image enhancement using adaptive filtering for enhanced sift-based image matching. *Journal of Theoretical and Applied Information Technology*. 2013, vol. 51, pp. 392–399.

64. ANCUTI, Codruta; ANCUTI, Cosmin; VLEESCHOUWER, Christophe De; GARCIA, Rafael. Locally Adaptive Color Correction for Underwater Image Dehazing and Matching. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2017, pp. 997–1005. ISSN 2160-7516. Available from DOI: `10.1109/CVPRW.2017.136`.

65. AGRAFIOTIS, Panagiotis; DRAKONAKIS, Georgios I.; GEORGOPOULOS, Andreas; SKARLATOS, Dimitrios. The Effect Of Underwater Imagery Radiometry On 3D Reconstruction And Orthoimagery. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. 2017, vol. XLII-2/W3, pp. 25–31. Available from DOI: `10.5194/isprs-archives-XLII-2-W3-25-2017`.

66. MANGERUGA, Marino; BRUNO, Fabio; COZZA, Marco; AGRAFIOTIS, Panagiotis; SKARLATOS, Dimitrios. Guidelines for Underwater Image Enhancement Based on Benchmarking of Different Methods. *Remote Sensing*. 2018, vol. 10, no. 10. ISSN 2072-4292. Available from DOI: `10.3390/rs10101652`.

67. TAKETOMI, Takafumi; UCHIYAMA, Hideaki; IKEDA, Sei. Visual SLAM algorithms: a survey from 2010 to 2016. *IPSJ Transactions on Computer Vision and Applications*. 2017, vol. 9, no. 1. ISSN 1882-6695. Available from DOI: `10.1186/s41074-017-0027-2`.

68. YOUNES, Georges; ASMAR, Daniel; SHAMMAS, Elie; ZELEK, John. Keyframe-based monocular SLAM: design, survey, and future directions. *Robotics and Autonomous Systems*. 2017, vol. 98, pp. 67–88. ISSN 0921-8890. Available from DOI: `https://doi.org/10.1016/j.robot.2017.09.010`.

69. KLEIN, Georg; MURRAY, David. Parallel Tracking and Mapping for Small AR Workspaces. In: *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*. 2007, pp. 225–234. Available from DOI: `10.1109/ISMAR.2007.4538852`.

70. MUR-ARTAL, Raúl; MONTIEL, Jose Maria Martínez; TARDÓS, Juan D. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*. 2015, vol. 31, no. 5, pp. 1147–1163. ISSN 1552-3098. Available from DOI: 10.1109/TRO.2015.2463671.

71. SCHÖPS, Thomas; ENGEL, Jakob; CREMERS, Daniel. Semi-dense visual odometry for AR on a smartphone. In: *2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 2014, pp. 145–150. Available from DOI: 10.1109/ISMAR.2014.6948420.

72. ENGEL, Jakob; SCHÖPS, Thomas; CREMERS, Daniel. LSD-SLAM: Large-Scale Direct Monocular SLAM. In: FLEET, David; PAJDLA, Tomas; SCHIELE, Bernt; TUYTELAARS, Tinne (eds.). *Computer Vision – ECCV 2014*. Cham: Springer International Publishing, 2014, pp. 834–849. ISBN 978-3-319-10605-2.

73. KÄHLER, Olaf; PRISACARIU, Victor Adrian; REN, Carl Yuheng; SUN, Xin; TORR, Philip; MURRAY, David. Very High Frame Rate Volumetric Integration of Depth Images on Mobile Devices. *IEEE Transactions on Visualization and Computer Graphics*. 2015, vol. 21, no. 11, pp. 1241–1250. ISSN 1077-2626. Available from DOI: 10.1109/TVCG.2015.2459891.

74. PIAO, Jin-Chun; KIM, Shin-Dug. Adaptive Monocular Visual–Inertial SLAM for Real-Time Augmented Reality Applications in Mobile Devices. *Sensors*. 2017, vol. 17, no. 11. ISSN 1424-8220. Available from DOI: 10.3390/s17112567.

75. LI, Peiliang; QIN, Tong; HU, Botao; ZHU, Fengyuan; SHEN, Shaojie. Monocular Visual-Inertial State Estimation for Mobile Augmented Reality. In: *2017 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 2017, pp. 11–21. Available from DOI: 10.1109/ISMAR.2017.18.

76. LUCAS, Bruce David; KANADE, Takeo. An Iterative Image Registration Technique with an Application to Stereo Vision. In: *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*. Vancouver, BC, Canada: Morgan Kaufmann Publishers Inc., 1981, pp. 674–679. IJCAI'81.

77. SHI, Jianbo; TOMASI, Carlo. Good features to track. In: *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 1994, pp. 593–600. ISSN 1063-6919. Available from DOI: 10.1109/CVPR.1994.323794.

78. MUÑOZ-SALINAS, Rafael; MARÍN-JIMÉNEZ, Manuel Jesús; YEGUAS-BOLIVAR, Enrique; MEDINA-CARNICER, Rafael. Mapping and localization from planar markers. *Pattern Recognition*. 2018, vol. 73, pp. 158–171. ISSN 0031-3203. Available from DOI: https://doi.org/10.1016/j.patcog.2017.08.010.

79. GUTH, Felipe; SILVEIRA, Luan; BOTELHO, Silvia; DREWS, Paulo; BALLESTER, Pedro. Underwater SLAM: Challenges, state of the art, algorithms and a new biologically-inspired approach. In: *5th IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics*. 2014, pp. 981–986. ISSN 2155-1774. Available from DOI: 10.1109/BIOROB.2014.6913908.

80. HIDALGO, Franco; BRÄUNL, Thomas. Review of underwater SLAM techniques. In: *2015 6th International Conference on Automation, Robotics and Applications (ICARA)*. 2015, pp. 306–311. Available from DOI: 10.1109/ICARA.2015.7081165.

81. FERREIRA, Fausto; MACHADO, Diogo; FERRI, Gabriele; DUGELAY, Samantha; POTTER, John. Underwater optical and acoustic imaging: A time for fusion? a brief overview of the state-of-the-art. In: *OCEANS 2016 MTS/IEEE Monterey*. 2016, pp. 1–6. Available from DOI: 10.1109/OCEANS.2016.7761354.

82. FERREIRA, Fausto; VERUGGIO, Gianmarco; CACCIA, Massimo; BRUZZONE, Gabriele. Real-time optical SLAM-based mosaicking for unmanned underwater vehicles. *Intelligent Service Robotics*. 2012, vol. 5, pp. 55–71. ISSN 1861-2784. Available from DOI: 10.1007/s11370-011-0103-x.

83. ZHANG, Jun; ILA, Viorela; KNEIP, Laurent. Robust Visual Odometry in Underwater Environment. In: *2018 OCEANS - MTS/IEEE Kobe Techno-Oceans (OTO)*. 2018, pp. 1–9. Available from DOI: 10.1109/OCEANSKOBE.2018.8559452.

84. GALLAGHER, Dennis G. Development of Miniature, Head-Mounted, Virtual Image Displays for Navy Divers. In: *OCEANS '99*. 1999, vol. 3, pp. 1098–1104. Available from DOI: 10.1109/OCEANS.1999.800143.

85. MORALES, Rogelio; KEITLER, Peter; MAIER, Patrick; KLINKER, Gudrun. An Underwater Augmented Reality System for Commercial Diving Operations. In: *OCEANS 2009*. 2009, pp. 1–8. ISSN 0197-7385. Available from DOI: 10.23919/OCEANS.2009.5422365.

86. BROWN, Hunter C.; WANG, Haozhu. Underwater Augmented Reality: Navigation and identification. In: *2013 OCEANS - San Diego*. 2013, pp. 1–5. ISSN 0197-7385. Available from DOI: 10.23919/OCEANS.2013.6740996.

87. BLUM, Lisa; BROLL, Wolfgang; MÜLLER, Stefan. Augmented Reality Under Water. In: *SIGGRAPH '09: Posters*. New York, NY, USA: ACM, 2009. Available from DOI: 10.1145/1599301.1599398.

88. OPPERMANN, Leif; BLUM, Lisa; LEE, Jun-Yeong; SEO, Jung-Hyub. AREEF Multi-player Underwater Augmented Reality experience. In: *2013 IEEE International Games Innovation Conference (IGIC)*. 2013, pp. 199–202. Available from DOI: 10.1109/IGIC.2013.6659137.

89. OPPERMANN, Leif; BLUM, Lisa; SHEKOW, Marius. Playing on AREEF: Evaluation of an Underwater Augmented Reality Game for Kids. In: *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services*. New York, NY, USA: ACM, 2016, pp. 330–340. Available from DOI: 10.1145/2935334.2935368.

90. BELLARBI, Abdelkader; DOMINGUES, Christophe; OTMANE, Samir; BENBELKACEM, Samir; DINIS, Alain. Augmented reality for underwater activities with the use of the DOLPHYN. In: *10th IEEE International Conference on Networking, Sensing and Control (ICNSC)*. 2013, pp. 409–412. Available from DOI: 10.1109/ICNSC.2013.6548773.

91. GETREUER, Pascal. Automatic Color Enhancement (ACE) and its Fast Implementation. *Image Processing On Line*. 2012, vol. 2, pp. 266–277. Available from DOI: 10.5201/ipol.2012.g-ace.

92. MOREL, Jean-Michel; PETRO, Ana-Belen; SBERT, Catalina. Screened Poisson Equation for Image Contrast Enhancement. *Image Processing On Line*. 2014, vol. 4, pp. 16–29. Available from DOI: 10.5201/ipol.2014.84.

93. LAVEST, Jean-Marc; RIVES, Gérard; LAPRESTÉ, Jean-Thierry. Dry camera calibration for underwater applications. *Machine Vision and Applications*. 2003, vol. 13, no. 5, pp. 245–253. ISSN 1432-1769. Available from DOI: 10.1007/s00138-002-0112-z.

Development and integration of digital technologies addressed to raise awareness and access to European underwater cultural heritage. An overview of the H2020 i-MARECULTURE project

Fabio Bruno, Antonio Lagudi, Gerardo Ritacco
3D Research s.r.l. – University of Calabria
Rende (CS), Italy
(fabio.bruno, antonio.lagudi, gerardo. ritacco)@unical.it

Panagiotis Agrafiotis, Dimitrios Skarlatos
Photogrammetric Vision Lab
Cyprus University of Technology
Cyprus
(panagiotis.agrafioti, dimitrios.skarlatos)@cut.ac.cy

Jan Čejka, Pavel Kouřil, Fotis Liarokapis
HCI Lab, Faculty of Informatics
Masaryk University
Brno, Czech Republic
(396550, 324987, liarokap)@mail.muni.cz

Oliver Philpin-Briscoe, Charalambos Poullis, Sudhir Mudur, Bart Simon
Department of Computer Science and Software Engineering Concordia University
Montréal, Canada
charalambos@poullis.org

***Abstract*—The Underwater Cultural Heritage (UCH) represents a vast historical and scientific resource that, often, is not accessible to the general public due the environment and depth where it is located. Digital technologies (Virtual Museums, Virtual Guides and Virtual Reconstruction of Cultural Heritage) provide a unique opportunity for digital accessibility to both scholars and general public, interested in having a better grasp of underwater sites and maritime archaeology. This paper presents the architecture and the first results of the Horizon 2020 i-MARECULTURE (Advanced VR, iMmersive Serious Games and Augmented REality as Tools to Raise Awareness and Access to European Underwater CULTURal heritage) project that aims to develop and integrate digital technologies for supporting the wide public in acquiring knowledge about UCH. A Virtual Reality (VR) system will be developed to allow users to visit the underwater sites through the use of Head Mounted Displays (HMDs) or digital holographic screens. Two serious games will be implemented for supporting the understanding of the ancient Mediterranean seafaring and the underwater archaeological excavations. An Augmented Reality (AR) system based on an underwater tablet will be developed to serve as virtual guide for divers that visit the underwater archaeological sites.**

Keywords— Underwater Archaeological sites, Virtual museums, Serious games, Underwater Augmented Reality system.

I. INTRODUCTION

Underwater archaeological assets represent a relevant part of the world cultural heritage and a particularly important element in the history of people, nations, and their relation with each other concerning their common heritage, especially in the European context. For this reason, the i-MARECULTURE (Advanced VR, iMmersive Serious Games and Augmented REality as Tools to Raise Awareness and Access to European Underwater CULTURal heritage) project aims to enhance the visit experience in museums and underwater parks by developing new technological solutions able to stimulate the public in increasing their knowledge about the European maritime archaeology.

The use of new technologies for improving the exploitation of Underwater Cultural Heritage (UCH) started ten years ago with the VENUS (Virtual exploration of underwater site) project [1] that strongly focused on the virtual reconstruction of underwater archeological sites and several campaigns have been conducted to acquire detailed 3D models of ancient shipwrecks. The digital models of the underwater sites have been used in a couple of Virtual Reality (VR) and Augmented Reality (AR) tools for interactive and immersive visualization [2], allowing archaeologists to study the virtual site from within. More recent works have proposed different frameworks for the collection and visualization of the underwater cultural assets by means of VR technologies, but these results limit the exploitation to a single underwater archaeological remain [3] or are more oriented to the digitization for scientific purposes [4], rather than focusing on edutainment for general audiences. Some VR based demonstrators have also been designed for the general public, but the virtual exhibit is more oriented to the presentation and visualization of archaeological data [5] rather than to edutainment purposes.

One of the few works related to the adoption of an edutainment approach for supporting the knowledge acquisition about UCH has been described by Stone [6] who has applied the

serious game approach in order to raise the awareness of schoolchildren and public audiences to the importance of protecting global oceanic resources. In particular, the subsea environment, in which the wreck of the naval vessel Scylla lies, has been represented to visually simulate the life cycles of dynamic sea life forms and colonization processes in order to increase the cultural awareness about marine biology rather than underwater archaeology.

Recently, the VISAS (VIrtual and augmented exploitation of Submerged Archaeological Sites) project [7] proposed a virtual diving system based on a VR application consisting in the simulation of a real diving session from the point of view of a scuba diver. The application follows a storyline described by a virtual diving companion who guides users during the exploration of the underwater archaeological site. The virtual diving system provides general and historical-cultural contents, but also information about the flora and fauna of the specific submerged site to the users.

Compared to previous works, the i-MARECULTURE project follows an interdisciplinary approach that goes beyond the mere technological development. The challenge is to merge existing upcoming technologies, adjust them to the underwater environment, in order to bring inherently unreachable European UCH, within reach and understanding of general public. The final aim is the development of personalized experience, using several methods to provide digital content, before, during and after a visit to an archaeological park or a museum.

II. PROJECT OVERVIEW

To raise awareness and access to otherwise unreachable (or difficult to reach) European UCH, the project utilizes VR and AR technologies to prologue and personalize virtual and actual visits to submerged shipwreck and other archaeological sites. Therefore, implements a serious seafaring game to raise awareness even before public visits maritime museums or sites. In the museum, the visitor will have the opportunity to have a personalized dry visit to project's test sites, using equipment which is unavailable to wide public due to high cost, such as Holographic screens and high end VR Head Mount Displays (HMDs). During the virtual visits he will have the ability to follow his own path and acquired details about certain objects or details that he selects. If the visit is taken in a submerged site, the diver-visitor will be equipped with an AR enabled underwater tablet. After the museum visit, the visitors who wish to extend the visit, will have the ability to make an online dry visit from his home, using VR HMD, although in a reduced detail and resolution virtual world. An additional VR game, which we will be able to play from home, will provide even more information about the difficulties of the underwater documentation and excavation of such sites.

It should be mentioned that the experienced maritime archaeologists of the project, ensure the historic accuracy of every aspect of i-MARECULTURE project, both through storytelling as well as realistic representations and game scenarios.

Regarding the dry VR visits, the focus will be given to the quality of immersion into the virtual world, using as much detail as possible and high end equipment. The AR tablet will provide navigation guidance to the diver/visitor and additional information about specific exhibits based on proximity and diver's preferences. If it will be used to explore a sunken city, the AR tablet will provide the diver with conceptual designs of the underwater villas and related constructions, which are now laying in ruins. Statues that have been removed for conservation and preservation reasons will be displayed in their original position and state in the tablet's screen, overlaid on the actual video recording of the camera in real time.

The seafaring game will be used to attract and familiarize the public with ancient maritime trade and commerce, between ports and civilizations of the eastern Mediterranean during Classic and Hellenistic period. The user as a captain of an ancient ship, must sail across ports trying to increase his fortune, while staying alive and overcoming all difficulties of those trips. At the same time the user is introduced with proper storytelling to support background knowledge about ports, weather, risks, routes, cargo load, commodities, and crew, necessary to make correct decisions.

The VR game is focusing in the challenge of underwater excavation in general and digging in particular. The player will be introduced in all phases of an ancient underwater shipwreck discovery and proper actions until he will be able to participate in a digging dive, to expose and recover artifacts. Storytelling and game plot will ensure that the game relays the proper message to players regarding all necessary actions preceding an archaeological excavation. This game will even have the option of multi-player collaboration and will also be used as training for future maritime archaeologists. The shipwreck site will be recreated procedurally each time the player starts a game, to ensure unpredictability.

In the underwater serious game, storytelling is not as evident as in Seafaring serious game. Still, will be created an intro story which will introduce the users with game topic, aim and structure. This intro should offer enough information, but has to be short and attract users to continue and play the game. An archaeologist character could be a storyteller in this game as well, because of the positive impact of such storytelling technique on users.

A by-product of i-MARECULTURE project are two libraries of 3D ship models and amphorae of that period. The 3D objects of these libraries will be used in the two serious games. Users will be given the ability to select ship and shipwreck type they wish to play with. The libraries are being designed with ontology, so that they can be useful for further archaeological archiving and will be released to public.

A. Pilot sites

Three different pilot sites have been selected for testing and demonstration activities. The selected sites are representative of different kind of UCH, of different states of environmental and geomorphologic conditions (i.e. water depth, water turbidity etc.) and of different periods, in order to present the users a wide range of the common European maritime culture.

The first site is the Underwater Archaeological Park of Baiae located off the north-western coasts of the bay of Puteoli (Naples). This site is part of the coastal region known as Campi Flegrei, that has been characterized by a periodic volcanic and

hydrothermal activity and it has been subjected to bradyseism, namely gradual changes in the levels of the coast with respect to the sea level (Fig. 1).

Fig. 1. Aerial photography of the Underwater Archaeological Park of Baiae (Italy).

The Park, which has an area of about 176.6 hectares safeguards the archaeological remains of the Roman city and the infrastructures of the roman harbor named Portus Iulius. Since antiquity this coastal region has been subject to the phenomenon of bradyseism, which may be positive or negative, and in its present state, the remains of the Roman Era are submerged at a depth ranging between 1 and 14-15m below sea level. The ancient Baiae was a famous seaside town much prized in antiquity for its temperate climate, beautiful setting and the properties of its mineral waters that have been exploited since the second century BC. It was the one of the most popular resort among the Roman aristocracy and the Imperial family up until the end of the fourth century A.D., when several ground movements caused the submersion of the city. The area selected as Pilot site is the complex of the "Villa con ingresso a protiro" located at 5/6 meters' depth. The rooms which composed the Villa extend for 40 meters on the road flanked by thermae, tabernae and other villas; however, its real size could be larger. The data available for the site are digital photos, drawing and 3D models created by the Underwater Archaeology Unit of the Istituto Superiore per la Conservazione ed il Restauro directed by Barbara Davidde [8].

The second site is the Mazotos shipwreck that lies at a depth of 44 m, ca.14 nautical miles (NM) southwest of Larnaca, Cyprus, off the coast of Mazotos village, 1.5 NM from the shore (Fig. 2).

Fig. 2. Mazotos shipwreck, Cyprus.

The wreck lies on a sandy, almost flat seabed and consists of an oblong concentration of at least 800 amphorae, partly or totally visible before any excavation took place. This assemblage appears to be nearly in the form of a ship, with its south end pointed and the north one almost squared off. It measures approximately 1 m at its maximum vertical relief, 16 m on its long axes and 6.5 m on its maximum width [9, 10].

The shipwreck was reported to the Department of Antiquities in 2006. The investigation of the shipwreck is conducted jointly by the Maritime Research Laboratory (MARE Lab) of the University of Cyprus and the Department of Antiquities, under the direction of Dr. Stella Demesticha. It is the first shipwreck of the late classical period, found in the Southeast Mediterranean carrying Chian amphorae, but its significance lies mainly in its excellent state of preservation. The Department of Civil Engineering and Geomatics at the Cyprus University of Technology (CUT), under the direction of Dr. Dimitrios Skarlatos, and the MARE Lab, under the direction of Dr. Stella Demesticha, have been surveyed the Mazotos site several times thus producing a complete documentation of the site, which includes the point clouds, database, photographs of finds, 3D solid models of artefacts and the 3D Site Model [12]. A combination of computer vision and photogrammetric methods has been used as a rapid means to obtain measurements with surveying accuracy in the scale of 1-2cm.

The third site is the Xlendi shipwreck that was named after the place where it was found off the Gozo coast in Malta (Fig. 3).

Fig. 3. Xlendi shipwreck, Gozo (Malta).

The site is curated by Timmy Gambin (University of Malta) and studied by himself. The wreck was on the focus of the GROPLAN project (http://www.groplan.eu) where Timmy Gambin and Jean-Christophe Sourisseau (Aix-Marseille University) have studied this wreck. The shipwreck was detected by Aurora Trust, an American consortium expert in deep-sea detection systems, during a prospecting campaign in 2008. The inspection operation was authorized and mandated by Heritage Malta and Malta's "Superintendence of Cultural Heritage" with the aim of taking an inventory of all the underwater ruins located in Malta's territorial water at a depth between 50 and 150 meters. A preliminary expertise helped to characterize a very ancient ship, probably the oldest discovered in the western Mediterranean. An intervention group was gradually created by Timmy Gambin in order to put together a strategy adapted to the study of this shipwreck. The shipwreck

is located near a coastline known for its limestone cliffs that plunge into the sea and whose foundation rests on a continental shelf at an average depth of 100 m below sea level. The shipwreck rests on a practically flat area of this submerged plateau at a depth of 94 m. Sonar images as well as the image of the entire deposit site shows that the cargo remains very closely grouped together and appears to have been very slightly disturbed.

A photogrammetric survey was done in 2014 from the submarine Remora 2000 by COMEX and LSIS CNRS - AMU in the framework of the GROPLAN project. As a first result a dense cloud of point, scaled, is available with a mesh, covering the whole site, including amphora moved far from the wreck. In addition, a very high resolution orthophoto was computed (41507 x 60377 pixels with resolution = 0.005 mm / Pixel) and accessible from a web site.

III. Data gathering and preparation

Several underwater 3D acquisition methods are available nowadays, but for detailed archaeological documentation photogrammetry is the most prominent [13]. Except of image based only techniques, other methods exploiting active sensors are used for detailed underwater archaeological 3D recording. These methods include Time-of-Flight (ToF) or LiDAR techniques [8], triangulation scanners and structured light systems. The ToF 3D scanners obtain range information by measuring the time delay of a light signal between the system and the object. Triangulation scanners project a light beam to measure the direction of light from a known position and having a well-known geometric configuration. Regarding underwater structured light systems, in [3] is described a system consisting of two commercial optical cameras and a digital projector. Finally, acoustic systems use sound propagation properties to record underwater objects by calculating the distance to the object using the time of travel of the sound. The most common used are the Multibeam Echo sounders (MBEs), the Side Scan Sonar (SSS) and the Sub Bottom Profiler (SBP). These systems are usually mounted in a moving platform such as a Remotely Operated Vehicle (ROV) or an Autonomous Underwater Vehicle (AUV), a vessel's hull etc. Compared to optical systems, the acoustic ones provide less spatial resolution but longer range.

Since the combination of computer vision algorithm with photogrammetric techniques, image based modelling transformed from an exotic expert-only method to a trivial field task for the average archaeologist scholar. Nevertheless, although to the reach of the wide public, several limiting factor and problems in the underwater environment, pose challenges even to the experts. Given that photogrammetric accuracy depends mainly to object to camera distance, extremely detailed 3D models may be acquired underwater. This is an invaluable characteristic of photogrammetry for underwater 3D modelling.

Several examples of shipwreck documentation [11, 14] and sites [15] are to confirm this argument. All three sites involved in the i-MARECULTURE project have been detailed documented using underwater photogrammetry [10, 11, 16, 17], Structure from Motion (SfM) and Multi View Stereo (MVS) techniques. In all cases strobes and underwater lights have been used either because of the depth (Xlendi and Mazotos sites) or due to poor lighting conditions is certain areas to the shallow Baiae site. The 3D textured models have been generated from dense point clouds and are extremely detailed, appropriate to be used for virtual reality and dry visits (Fig. 4).

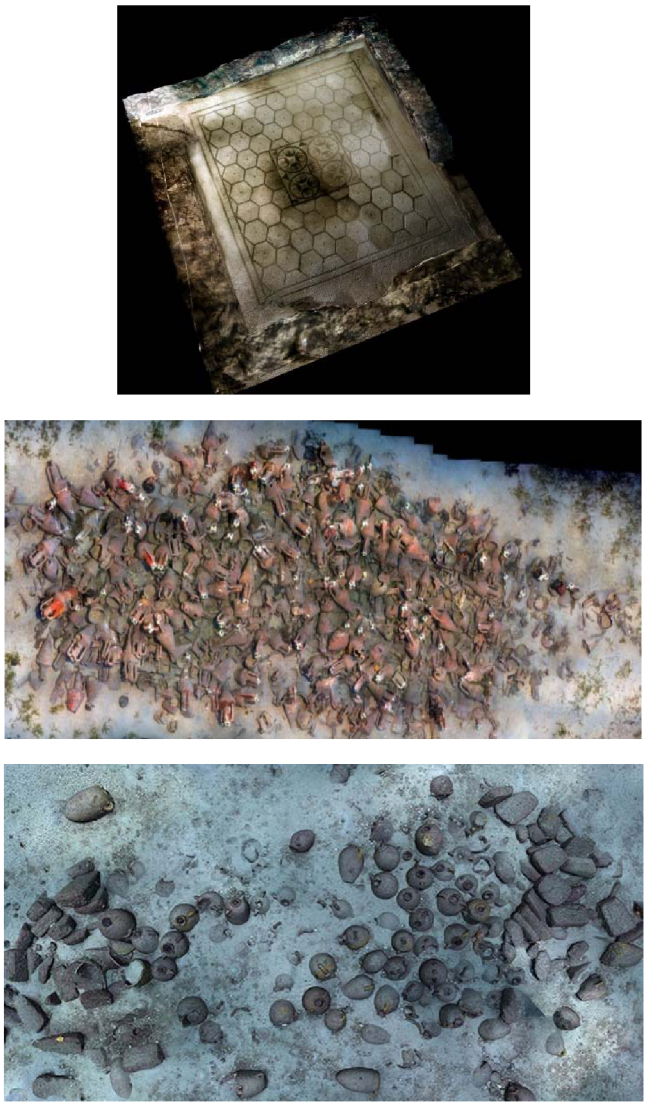


Fig. 4. 3D reconstructions of the i-MARECULTURE pilot sites: A portion of the "Villa con ingresso a protiro" at Baiae Park (top), Mazotos shipwreck (center), Xlendi shipwreck (bottom).

IV. A serious game for understanding ancient seafering in the Mediterranean Sea

To better understand ancient seafaring practices, we proposed the design and development of a serious seafaring game. The game's users will be able to experience the role of the ship captain/merchant. The objective of the game will be to trade goods, complete quests and face off the many perils of the high seas in a quest to uncover new sources of wealth and opportunity in distant lands based on the information we have and hypothesize for the roman-hellenistic era.

The game will make use of the probabilistic geospatial analysis of the ship routes of the Classical and Hellenistic

period through the re-use and spatial analysis from open GIS maritime data, ocean and weather data. Naval engineering and sailing techniques along with the aforementioned ship routes, are used as underlying information for the seafaring game.

A game map is created as the route-plotting graph shown in Fig. 5, where each vertex represents a coastal feature, i.e. port, and the edges represent “hops” in a voyage.

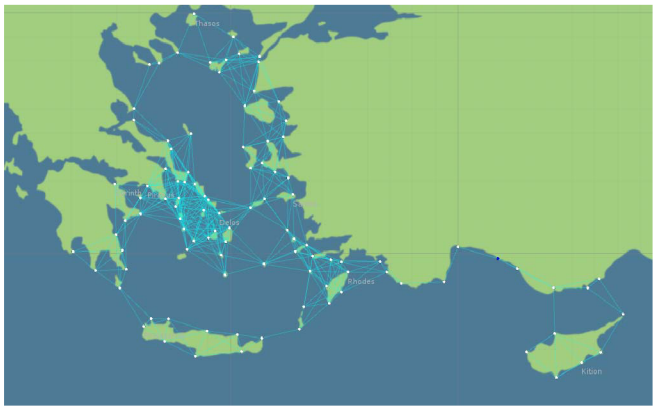


Fig. 5. Route creation algorithm in the game map.

Currently, the cost function is the edge length, and the heuristic is the distance to the goal vertex. This is sufficient for estimating voyage time, but additional hazard data (piracy, weather, shallow waters, etc.) add depth to the simulated economy by exerting negative incentive along dangerous routes. This could result in high-risk destinations where certain resources are scarce, offering the player a large reward due to the correspondingly high demand. Hazard data can be defined discretely for each edge, or sampled from a heat map.

The information provided by the archaeologists has been incorporated into the game’s project data e.g. port types, commodity types.

Lastly, path-finding (A*) and spline (Catmull-Rom) tools were required to implement the route finding and route display features. We are currently in the first steps of evaluating the game with end-users.

V. VR TECHNOLOGIES FOR UNDERWATER EXPLORATION AND EXCAVATION

A. *Virtual Underwater Exploration*

The technological requirements for applying serious games to cultural heritage have been previously explored and the most characteristic case studies were presented [18]. In particular, the case studies have been categorized into three types of computer-game-like applications, namely a) prototypes and demonstrators, b) virtual museums, and c) commercial historical games. The state-of-the-art in Serious Game technology is identical to the state-of-the-art in Entertainment Games technology. Meaning gamification techniques will inform how information should be represented for the different users of the platform and adaptivity will be experimented on. Our approach will exploit the interactive characteristics of VR games.

The aim of the serious game is to raise people's archaeological knowledge and cultural awareness. It will provide immersive technologies to increase interaction time in an underwater archaeological site, both for the public, as well as, for researchers and scholars. Users can experience an immersive virtual underwater visit using off-the-shelf VR headsets (i.e. HTC Vive). Apart from their visit, they can also get some information about the archaeological artefacts (i.e. textual descriptions, videos and sounds).

The application was developed in Unity engine with support for VR Head Mounted Displays (HMD) HTC Vive. Apart from the HMD (resolution per eye 1080 x 1200, refresh rate 90 Hz, field of view 110°), HTC Vive comes bundled with two motion tracked controllers and laser based tracking system called Lighthouse providing 6 Degrees-of-Freedom (DOF) tracking in an up to 4.5 x 4.5 m area with two beacons. This setup, allows users to perceive a very immersive experience of the underwater environment. Work done for developing the immersive VR environment consists of four main parts [19]: a) graphics effects (i.e. lightning, fog), b) procedural content (i.e. placement of artefacts), c) behavior (i.e. fish movement) and d) interaction (i.e. exploring and learning). Fig. 6 provides an example of fish behavior.

Fig. 6. Fish in the underwater immersive experience.

In the virtual experience, there are two types of fish which have different sizes and speeds, spawning groups of 80 and 100 instances. Fish school simulation implementation was based on the algorithm proposed in [20]. The algorithm simulates flocking behavior boids (bird-oid objects) - birds and other living species, including fish.

The user interaction is focused on navigating inside the virtual environment and receiving information about the archaeological artefacts Fig. 7. The user can either physically walk within a space of 3x3 meters or 'teleport' in the virtual space. Using the controls of the HTC Vive device, one can select the teleporting destination. This allows the user to 'jump' from one virtual position to another without physically moving. Players can interact directly in the virtual visits with the environmental objects (i.e. game objects). If appropriate, players can pick them up and examine them using the dedicated controllers. They can also control the relevant information that they will receive. For example, they can receive textual

descriptions about an artefact and then watch a relevant video (by pressing the touchpad of the right HTC Vive controller).

Fig. 7. User interaction in the underwater virtual exploration.

Another way to provide immersive experience to museum visitors is obtained by using holographic displays. In particular, we use the HoloVizio screens, developed by Holografika [21]. Using this technology, no additional gears are required and content observation depends on the user location and direction as walking in front of the display. It allows a large number of freely-moving simultaneous viewers to share the same underwater scene, viewing the 3D captured artefacts and seeing their different details with high quality. Visitors can even "look behind" the displayed objects, similarly to the real scene, having a true spatial feeling.

The approach used by HoloVizio technology is quite different from that of stereoscopic, multiview, volumetric and holographic systems. It uses a specially arranged array of projection modules and a holographic screen. The light beams generated in the projection modules hit the screen points in various angles and the holographic screen makes the necessary optical transformation to compose these beams into a perfectly continuous 3D view. With proper software control, light beams leaving the pixels propagate in multiple directions, as if they were emitted from the points of 3D objects at fixed spatial locations. In this way, viewers will perceive the points in objects behind the screen, or floating in the air in front of the screen, respectively. The HoloVizio displays generate the whole 3D light-field, not just limited numbers of views, and provide a continuous motion parallax in a wide Field of View (FOV). It is an ideal component of visualization systems to support the cooperative work of small research groups in archaeology presenting and assessing reconstruction hypothesis, or for large audiences sharing a common 3D view, as is the case of interactive installations in museums, where visual presentation can be supported without introducing wearable elements and complex management procedures.

B. Virtual Underwater Excavation

Another version of the underwater serious game is currently under development and will be primarily used for training maritime archaeology students. Users will be trained on the specifics of underwater excavation and familiarize themselves with the instruments in use, such as the airlift, without the constraints of the underwater environment, which, apart from the time limitations include the difficulty in verbal communication between the student and the instructor. Users will also be able to learn how to document and study wreck site formation processes. By teaming up with an expert, in this case a seasoned maritime archaeologist, students will gain valuable experience and knowledge on the methods, techniques and tools used in underwater archaeology, prior to working at the real site. The focus is not on simulating swimming but on excavating underwater following established archaeological methods and techniques.

For the "fun" element of the game, that motivates the user to play the game - we have chosen the treasure hunt gaming approach. The main idea is that one or more players who try to find hidden objects or places by following a series of clues (i.e. finding amphorae and another archaeological artifacts). To make the game more interesting, amphorae should be placed pseudo randomly, with some degree of covering by a sandy sea bed. The amphorae placement algorithm is based on the solution presented in [19] that places the objects procedurally. The placement of objects is different with every start of the game, however, there are fixed parameters - playing area size and number of amphorae. For our game, we have chosen 25 amphorae and playing area of 20 times 20 meters. The actual placement is done using an exponential function [19] (Fig. 8).

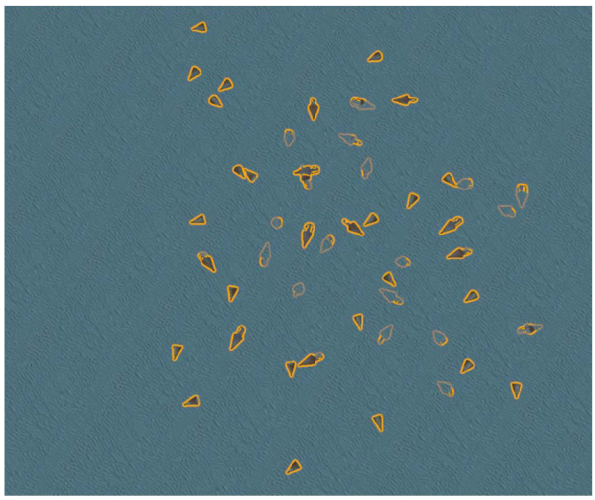


Fig. 8. Random object placement in the game scene.

An important element of the excavation game is that the objects should have different degree of coverage by the sandy terrain. Various methods for real-time modification of the terrain exists, which would allow the user to dig the objects. The methods differ on their computational difficulty and level of believability. One of the possible approaches, due to the limited amount of interaction the user is doing with the terrain (the digging is not a continuous operation done in every frame), modification of positions of the vertices of the mesh might be deployed to modify the terrain. However, more universal and correct approaches are available, e.g. the method introduced by Aquillo [22]. The method presents a real-time algorithm to

modify the Dynamically Displaced Height Map (DDHM) to record vehicle tracks in sandy terrain. The height map representation also allows us to use other algorithms based on height maps, which would add slippage properties to our soil [23] (Fig. 9).

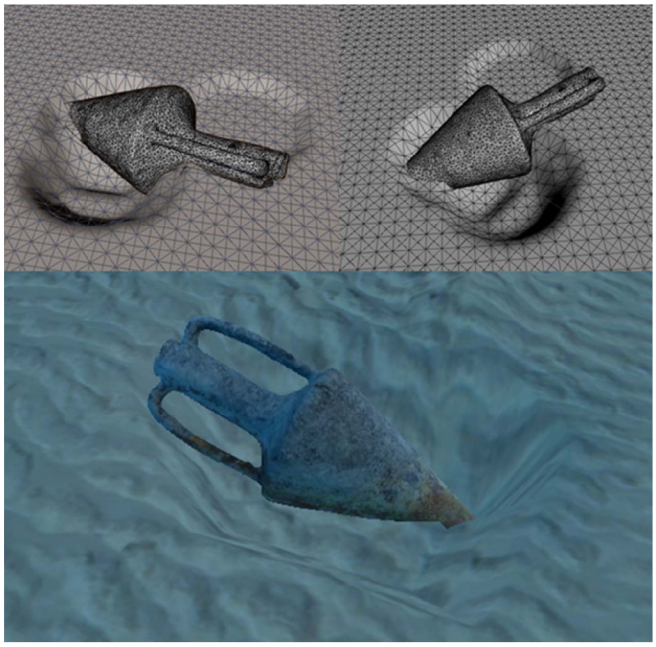


Fig. 9. Underwater digging in the serious game.

Moreover, the visual fidelity of the simulation could be increased by using a terrain-mesh blending techniques, to apply the textures of the sandy material to parts of the objects above the seabed.

VI. Uderwater Augmented Reality

The potential of AR to enhance immediate surroundings through the projection of digital content into users' real environment has been discussed in various research contexts [24 – 26]. For the tourism industry, an increasing number of scholars recognize the potential of AR for the enhancement of the tourism experience [27]. Using marker or location based AR applications, tourists can receive instant information on unknown surroundings that contribute to provide a new learning experience.

Systems for underwater AR have been already tested and used in the last years. HMDs have been used in [28] to help divers with orientation and navigation underwater by augmenting their vision with artificial horizon and visual aids to provide a guideline and help them with their work. An underwater device for AR that allows users to play simple games underwater has been introduced in [29]. The authors, in [30], use tablets placed into a waterproof case, and develop a game for kids that helps them to improve their swimming skills in a swimming pool.

So far, the only attempt to adopt AR technology directly in the submerged environment for the exploitation of UCH is represented by the VISAS project [7, 31]. The developed AR system consists in an underwater tablet equipped with an underwater positioning and orientation system that guides the diver tourists during the diving session while providing information about the archaeological artifacts spotted during the visit.

The technologies developed during the VISAS project will be used in i-MARECULTURE project for the implementation of an underwater AR application. It consists in an underwater tablet equipped with a hybrid tracking system that guides the diver during the exploration, while visualizing an AR layer that shows labels or 3D contents representing the current or the original aspect of the artefacts. The underwater AR interface will be capable of superimposing different types of visual information (i.e. 3D, metadata, images, videos and sound) over the images captured by the tablet main camera. The divers who visit the underwater archaeological site will have the possibility to see their position over the 3D bathymetry of the site and enjoy a hypothetical virtual reconstruction of the structures to better understand their original aspect and their function.

A. Underwater Augmented Reality Interfaces

The AR software running on the tablet is the way through which the diver will be able to live an augmented experience rather than a classic immersion. Thanks to a 3D map and the hybrid localization service, the software will help the diver to understand his position and orientation during exploration. The user interface (UI) of the AR software will be designed according to a user centered design (UCD) approach in order to make it easy to use and intuitive. The AR interface will superimpose an information layer over the images captured by the tablet main camera. The AR layer can include text, images, videos and 3D models of the reconstructed underwater site, or of single artefacts. This augmented information will help divers to better understand structural information of the archaeological site especially in the case of poor visibility conditions. The UI will provide also a command button that will allow users to switch from the 3D model of the archaeological remain to the 3D model of a hypothetical reconstruction of the artefact.

To increase the perception of the underwater scene, the AR application will allow to improve the vision of divers. Vision underwater is degraded by several factors, mostly by turbidity and absorption of lighting, which is different for different color channels. Concerning the turbidity in the underwater images, the problem is very similar to dehazing, i.e. to correct the effects of fog and haze in images taken above water. Many algorithms were developed to dehaze input images, for example a method described in [32], based on Dark Channel Prior, or a method presented in [33], based on color-lines. However, it is more challenging to apply them for removing turbidity because the characteristics of color images that they use are biased in the underwater images, due to uneven absorption of color channels.

To enhance the quality of underwater images, the authors in [34] use a combination of common techniques. They tested methods for color correction, white balancing, and histogram equalization. The solution described in [35] adapts Dark Channel Prior technique to work for underwater images by developing a separate method for each color channel. Some authors propose algorithms that dehaze images in real-time. A method that evaluates each pixel of the input image separately, and use CUDA to accelerate the computation is proposed in

[36]. In [37], the authors use SIMD instruction and OpenMP library to fully utilize a processor, and dehaze images in tens of milliseconds.

The workflow of the AR dehazing architecture is show in Fig. 10.

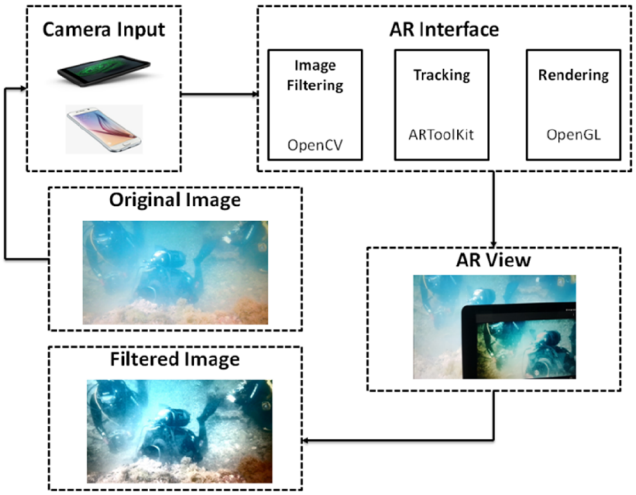


Fig. 10. Augmented reality dehazing architecture

In the first step, the input image coming from the tablet camera is filtered and improved to reduce defects and other imperfections caused by turbidity or poor lighting conditions. This preprocessed image serves in the second step as input for the detection of objects in real world that will be used by the AR tracking algorithm. In the third step, objects of augmented reality are composed together with improved preprocessed input image and rendered into output image. This image is then displayed on the screen of the tablet.

The AR dehazing architecture is implemented in Java and targets Android platforms. Input from the camera is in NV21 format, which is a variant of YUV420 format. We keep the image in this format, and in the preprocessing step, we apply Contrast Limited Adaptive Histogram Equalization (CLAHE) [38] to its Y channel, leaving U and V channels without any change. We use an implementation of CLAHE algorithm from OpenCV library [39]. The AR part is handled by ARToolKit library [40].

The AR dehazing solution was tested on several videos taken in underwater environment. The videos were played on a monitor of a PC, and a tablet was used to record, process and display the images, see Fig. 7. The resolution of input camera was 1280 x 720. We performed our tests on two devices, NVIDIA Shield K1 tablet and Samsung Galaxy S6 phone. The preprocessing step took 7.8 milliseconds on the tablet and 9.0 milliseconds on the phone, which in both cases allowed the application to present improved images in real-time.

Fig. 11. Augmented reality dehazing test results.

B. *Hybrid underwater tracking solution*

The implementation of the AR application requires the development of a solution to calculate the position and orientation of the tablet in the submerged environment. It is performed by means of a hybrid solution that merges data generated by visual tracking techniques (both marker and model based) with data from an acoustic modem, integrated with the underwater tablet, which estimates the position of the receiver by computing the distance from at least three fixed transmitters (beacons) placed on the seabed (LBL - Long Baseline technique). Moreover, data coming from an inertial platform and a depth sensor are used to improve the accuracy and increase the robustness in case of loss of signal from one or more beacons. The data coming from the various sensors are processed through data fusion and error estimation algorithms (Fig. 12).

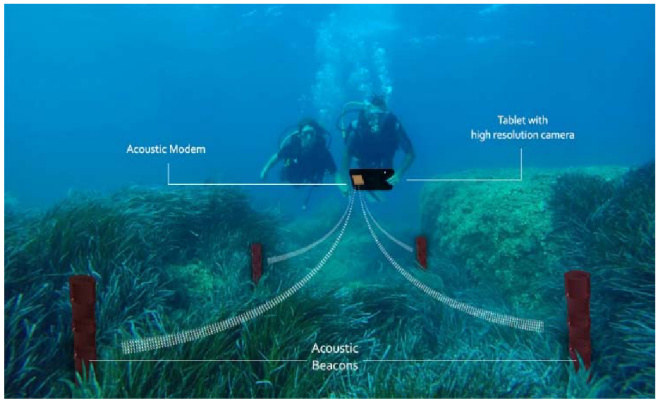


Fig. 12. Augmented Reality system.

Each beacon contains an electronic board to drive the piezoelectric transducer (Fig. 13, left). It operates at 25-30 KHz frequency band, greater than the audible band (0 - 20kHz), to avoid annoying sounds to be perceived by the divers during the

immersion. The beacon is equipped with a battery pack, which enables up to five hours of use, and with anchor rings used to fix it on the buoy cable. The beacons are positioned in known geographical points using a mooring post and a surface buoy that facilitates their recovery.

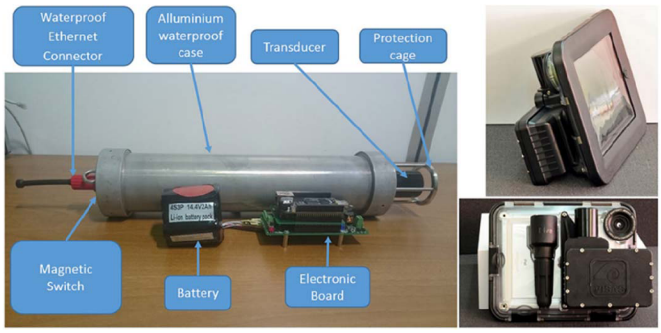


Fig. 13. VISAS beacon (left) and underwater tablet (right).

The tablet is composed of two major parts: a fully functional underwater touchscreen housing and a waterproof case for the electronics of the tracking system (Fig. 13, right). The two devices are connected using a Wi-Fi interface.

The tablet updates the diver's position and orientation on the 3D map at a frequency of 50 Hz. The current position is estimated through an Extended Kalman Filter that uses the distances from each beacon, the depth provided by a pressure sensor, the accelerations along the three axes (x, y, z), the orientations provided by an inertial platform and the position and orientation data provided by the visual odometry algorithm. The distance between the tablet and a beacon is computed using the time-of-fly of the acoustic wave. To this end, the acoustic modem on the tablet sends a two-way range (TWR) command to the remote beacon and starts an internal timer. When the tablet receives the response message sent by the beacon, the timer is stopped. The tracking system sends cyclically a TWR to all beacons. shows the localization algorithm.

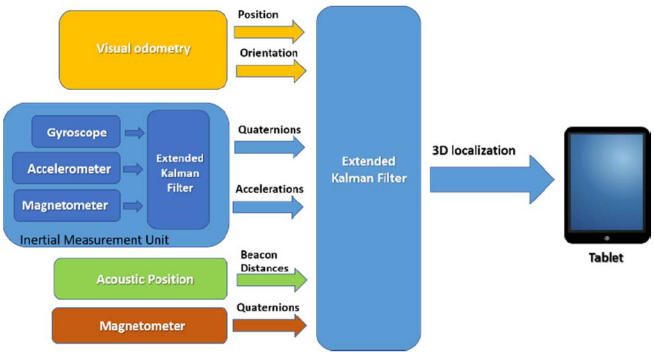


Fig. 14. Schematic of the localization algorithm.

VII. Conclusions

This paper has presented the i-MARECULTURE project and the partial results achieved so far. The project is characterized by the integration of different research activities in the areas of 3D acquisition, Virtual and Augmented reality, serious games, GIS, marine archaeology and storytelling. This synergy aims at creating innovative applications and digital experiences in the area of Virtual Museums in order to empower different types of users to engage with European underwater cultural heritage digital resources.

In particular, the paper has presented a VR application for underwater exploration and the first steps in the development of two serious games: the first one devoted to the simulation of ancient seafaring practice through the Mediterranean Sea and the second one for the simulation of underwater archaeological excavation. Moreover, the development of an underwater AR system has been presented. The system is based on an underwater tablet coupled with an acoustic localization device. The AR software will manage the image enhancement issue and will perform a hybrid localization by combining data generated by the acoustic device and by an optical tracking algorithm that is currently under development.

The tools will be validated and tested across real-world application coming from maritime archaeology research so as to achieve a great mixture between entertainment, informal educational, and underwater/maritime CH understanding.

Acknowledgment

The i-MARECULTURE has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 727153.

References

[1] P. Chapman, D. Roussel, P. Drap and M. Haydar, "Virtual Exploration of Underwater Archaeological Sites: Visualization and Interaction in Mixed Reality Environments", in *9th International Symposium on Virtual Reality, Archaeology and Cultural Heritage (VAST)*, pp. 141-148, 2008.

[2] P. Chapman et al., "Venus, virtual exploration of underwater sites", *Proceeding of joint event CIPA/VAST/EG/Euro-Med,* 2006.

[3] G. Varinlioğlu, "Data Collection for a Virtual Museum on the Underwater Survey at Kaş, Turkey", *International Journal of Nautical Archaeology,* vol. 40(1), pp. 182-188, 2011.

[4] I. Katsouri, A. Tzanavari, K. Herakleous, and C. Poullis, "Visualizing and assessing hypotheses for marine archaeology in a VR CAVE environment", *Journal on Computing and Cultural Heritage (JOCCH)*, vol. 8(2), p. 10. 2015.

[5] M. Haydar, D. Roussel, M. Maïdi, S. Otmane, and M. Mallem, "Virtual and augmented reality for cultural computing and heritage: a case study of virtual exploration of underwater archaeological site", *Virtual reality*, vol. 15(4), pp. 311-327, 2011.

[6] R. Stone, D. White, R. Guest, and B. Francis, "The Virtual Scylla: an exploration of serious games, artificial life and simulation complexity", *Virtual reality*, vol. 13(1), pp. 13-25, 2009.

[7] F. Bruno et al., "Project VISAS – Virtual and augmented exploitation of Submerged Archaeological Sites: overview and first results", *Marine Technology Society (MTS) Journal*, vol. 50(4), pp.119-129, 2016.

[8] B.D. Petriaggi, and G.G. de Ayala, "Laser scanner reliefs of selected archaeological structures in the submerged Baiae (Naples)", in *Intl. Arch. Photogramm. Remote Sens. Spatial Inf. Sci*., vol. 5, pp. 79–83, 2015.

[9] S. Demesticha, "The 4th - Century - BC Mazotos Shipwreck, Cyprus: a preliminary report", *International Journal of Nautical Archaeology*, vol. 40(1), pp. 39-59, 2011.

[10] S. Demesticha, D. Skarlatos, and A. Neophytou, "The 4th-century BC shipwreck at Mazotos, Cyprus: new techniques and methodologies in the 3D mapping of shipwreck excavations", *J. Field Archaeol*, vol. 39(2), pp. 134–150, 2014.

[11] D. Skarlatos, S. Demestiha, and S. Kiparissi, "An 'open' method for 3D modelling and mapping in underwater archaeological sites", *International Journal of Heritage in the digital era*, vol. 1(1), pp. 1-24, 2012.

[12] P. Drap, Underwater photogrammetry for archaeology, INTECH Open Access Publisher.

[13] F. Bruno, G. Bianco, M. Muzzupappa, S. Barone, and A.V. Razionale, “Experimentation of structured light and stereo vision for underwater 3D reconstruction”, *Journal of Photogrammetry and Remote Sensing*, ISPRS, vol. 66(4), pp. 508-518, 2011.

[14] P. Drap et al., “Underwater cartography for archaeology in the VENUS project”, *Geomatica*, vol. 62(4), pp. 419-427, 2008.

[15] J. Henderson, O. Pizarro, M. Johnson - Roberson, and I. Mahon, “Mapping Submerged Archaeological Sites using Stereo - Vision Photogrammetry”, *International Journal of Nautical Archaeology*, vol. 42(2), pp. 243-256, 2013.

[16] F. Bruno et al., “3D documentation of archeological remains in the underwater park of Baiae”, *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 40(5), p. 41, 2015.

[17] P. Drap et al., “Underwater photogrammetry and object modeling: a case study of Xlendi Wreck in Malta”, *Sensors*, vol. 15(12), pp. 30351-30384, 2015.

[18] E.F. Anderson et al. “Developing serious games for cultural heritage: a state-of-the-art review”, *Virtual reality*, vol. 14(4), pp. 255-275, 2010.

[19] F. Liarokapis et al., “3D Modelling and Mapping For Virtual Exploration of Underwater Archaeology Assets”, *Proc. of the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* (ISPRS), XLII-2/W3, Napflio, Greece, pp. 425-431, 2017.

[20] C.W. Reynolds, “Flocks, herds and schools: A distributed behavioral model”, *SIGGRAPH Comput. Graph.*, vol. 21(4), pp. 25–34, 1987.

[21] T. Balogh, P.T. Kovács, and Z. Megyesi, “Holovizio 3D display system”, in *Proceedings of the First International Conference on Immersive Telecommunications*, p. 19, ICST, October 2007.

[22] A. Aquilio, J. Brooks, Y. Zhu, and G. Owen, “Real-time GPU-based simulation of dynamic terrain”, *Advances in Visual Computing*, pp. 891-900, 2006.

[23] H. Prautzsch, A. Schmitt, J. Bender, and M. Teschner, “Soil deformation models for real-time simulation: a hybrid approach”, *Workshop on Virtual Reality Interaction and Physical Simulation VRIPHYS*, 2009.

[24] D.I. Han, T. Jung, and A. Gibson, “Dublin AR: Implementing Augmented Reality (AR) in Tourism”, In *Z. Xiang, & I. Tussyadiah (Eds), Information and Communication Technologies in Tourism*, pp. 511-523, Springer Computer Science: New York, 2013.

[25] M. Bordegoni et al. “Environment based on Augmented Reality and Interactive Simulation for Product Design Review”, in *Eurographics Italian Chapter Conference*, pp. 27-34, 2008.

[26] R. Hammady, M. Ma, and N. Temple N., “Augmented Reality and Gamification in Heritage Museums”, in *Joint International Conference on Serious Games*, Springer International Publishing, pp. 181-187, 2016.

[27] Leue M. C., Jung T., tom Dieck D. (2015) Google Glass Augmented Reality: Generic Learning Outcomes for Art Galleries. In I. Tussyadiah & A. Inversini (Eds.), Information and Communication Technologies in Tourism 2015 (pp. 463-476). Vienna:Springer.

[28] R. Morales, P. Keitler, P. Maier, and G. Klinker, "An Underwater Augmented Reality System for Commercial Diving Operations," in OCEANS 2009, MTS/IEEE Biloxi - Marine Technology for Our Future: Global and Local Challenges, 2009, pp. 1–8.

[29] A. Bellarbi, C. Domingues, S. Otmane, S. Benbelkacem, and A. Dinis, "Augmented reality for underwater activities with the use of the DOLPHYN," in 10th IEEE International Conference on Networking, Sensing and Control (ICNSC 2013), Evry, 2013, pp. 409-412.

[30] L. Oppermann, L. Blum, and M. Shekow, "Playing on AREEF: evaluation of an underwater augmented reality game for kids", in Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '16). ACM, New York, NY, USA, pp. 330-340.

[31] F. Bruno et al., “Virtual and Augmented Reality Tools to Improve the Exploitation of Underwater Archaeological Sites by Diver and Non-diver Tourists”, in *Euro-Mediterranean Conference*, Springer International Publishing, pp. 269-280, October 2016.

[32] K. He, J. Sun, and X. Tang, "Single Image Haze Removal Using Dark Channel Prior," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33 (12), pp. 2341-2353, 2011.

[33] R. Fattal, "Dehazing Using Color-Lines," in *ACM Transactions on Graphics*, vol. 34, pp. 1-14, 2014.

[34] C. Ancuti, C. O. Ancuti, T. Haber, and P. Bekaert, "Enhancing underwater images and videos by fusion," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 81-88, 2012.

[35] Y. Gao, H. Li, and S. Wen, "Restoration and Enhancement of Underwater Images Based on Bright Channel Prior," in *Mathematical Problems in Engineering*, p. 15, 2016.

[36] J. Zhang, and S. Hu, "A GPU-accelerated real-time single image de-hazing method using pixel-level optimal de-hazing criterion," in *Journal of Real-Time Image Processing*, vol. 9, pp. 661-672, 2014.

[37] J.-H. Kim, W.-D. Jang, J.-Y. Sim, and C.-S. Kim, "Optimized contrast enhancement for real-time image and video dehazing," in *Journal of Visual Communication and Image Representation*, vol. 24, pp. 410-425, 2013.

[38] S. M. Pizer et al., "Adaptive Histogram Equalization and Its Variations" in *Computer Vision, Graphics, and Image Processing*, vol. 39, pp. 355-368, 1987.

[39] Open Source Computer Vision Library. [Online]. Available: http://opencv.org.

[40] ARToolKit. [Online]. Available: https://artoolkit.org.

frontiers in Robotics and AI

ORIGINAL RESEARCH
published: 14 August 2018
doi: 10.3389/frobt.2018.00092

Impact of Dehazing on Underwater Marker Detection for Augmented Reality

Marek Žuži[1], Jan Čejka[1], Fabio Bruno[2], Dimitrios Skarlatos[3] and Fotis Liarokapis[1]*

[1] Human Computer Interaction Laboratory, Faculty of Informatics, Masaryk University, Brno, Czechia, [2] Department of Mechanical, Energy and Industrial Engineering, University of Calabria, Cosenza, Italy, [3] Photogrammetric Vision Laboratory, Department of Civil Engineering and Geomatics, Faculty of Engineering and Technology, Cyprus University of Technology, Limassol, Cyprus

Underwater augmented reality is a very challenging task and amongst several issues, one of the most crucial aspects involves real-time tracking. Particles present in water combined with the uneven absorption of light decrease the visibility in the underwater environment. Dehazing methods are used in many areas to improve the quality of digital image data that is degraded by the influence of the environment. This paper describes the visibility conditions affecting underwater scenes and shows existing dehazing techniques that successfully improve the quality of underwater images. Four underwater dehazing methods are selected for evaluation of their capability of improving the success of square marker detection in underwater videos. Two reviewed methods represent approaches of image restoration: Multi-Scale Fusion, and Bright Channel Prior. Another two methods evaluated, the Automatic Color Enhancement and the Screened Poisson Equation, are methods of image enhancement. The evaluation uses diverse test data set to evaluate different environmental conditions. Results of the evaluation show an increased number of successful marker detections in videos pre-processed by dehazing algorithms and evaluate the performance of each compared method. The Screened Poisson method performs slightly better to other methods across various tested environments, while Bright Channel Prior and Automatic Color Enhancement shows similarly positive results.

Keywords: dehazing, image restoration, underwater images, augmented reality, markers, tracking

OPEN ACCESS

Edited by:
Nikolaos Doulamis, National Technical University of Athens, Greece

Reviewed by:
John Quarles, University of Texas at San Antonio, United States
Athanasios Voulodimos, University of West Attica, Greece
Jian Chang, Bournemouth University, United Kingdom

****Correspondence:***
Marek Žuži
xzuzi@fi.muni.cz

Specialty section:
This article was submitted to Vision Systems Theory, Tools and Applications, a section of the journal Frontiers in Robotics and AI

Received: *27 April 2018*
Accepted: *24 July 2018*
Published: *14 August 2018*

Citation:
Žuži M, Čejka J, Bruno F, Skarlatos D and Liarokapis F (2018) Impact of Dehazing on Underwater Marker Detection for Augmented Reality. Front. Robot. AI 5:92. doi: 10.3389/frobt.2018.00092

1. INTRODUCTION

In the past several years, dehazing was gaining more attention in research as technology using digital imaging and computer vision is being used in more and more application domains. Dehazing has a special importance in applications where the images are degraded significantly by the environment. In scenes affected by haze, fog, or smoke, dehazing methods enhance or restore the quality of the images to make them usable in further processing.

Specific attention of scholars was recently drawn to the problem of underwater dehazing because digital imaging technology and computer vision became available for use under water in various areas such as underwater exploration, tracking, and 3D reconstruction of underwater objects, underwater archeology, or marine biology (Ludvigsen et al., 2007).

Physical properties of light propagation in water cause the underwater scenes to be affected by the absorption and scattering of light in the turbid medium. The light as an electromagnetic

radiation is significantly absorbed by the water as it propagates (Jerlov, 1976). Moreover, based on current turbidity conditions, water may contain a wide range of different solid particles that obstruct, scatter, or refract incident light. As a result, the images captured inside water suffer from low contrast, blur, color distortion, and uneven attenuation.

Therefore, dehazing methods developed for underwater scenes have to deal with the problems caused by the environment differently than the dehazing methods used in conventional scenes set in the atmosphere. Special methods for underwater image restoration or enhancement are developed to improve the visibility in images or videos created under water.

A technology that could benefit from dehazing methods capable of increasing the quality of underwater images is augmented reality (AR). As AR becomes popular and available in mobile devices, it is applied in various environments including scenes submerged in water. With devices capable of underwater AR, there is opportunity to create novel applications. A characteristic example would be maritime archaeology, and this is our ultimate goal. In such a case real-time reconstructions of ancient cities can be performed and it can be used by archaeologists in research as well as by tourists.

However, the image distortion created by water may affect the performance of AR applications negatively. Especially the visual tracking of objects in AR applications requires high quality of input data. If the quality of video captured for the AR application is decreased, the tracking may fail completely, making the rest of the application unable to augment the scene by additional information.

Understanding the way that hazy underwater environments affect the marker detection can help improve the performance of AR applications used underwater. This paper examines how dehazing can be used to improve the quality of underwater marker-based tracking for AR. First, well-known dehazing algorithms are evaluated in terms of improving the marker detection in different types of underwater scenes.

The possibility of improving the marker-based tracking underwater is shown by a comparison of several state-of-the-art dehazing methods developed for underwater environments. The methods were implemented and evaluated on a set of test videos showing various scenes. The evaluation considers four different underwater sites with different environmental conditions and aims to determine the effectiveness of selected dehazing algorithms to increase the success of marker detection.

Although AR scenarios require a high performance of used algorithms to ensure real-time responses, the focus of this paper is only to explore methods that can improve results of marker detection. The algorithms compared are not working in real-time performance and will have to be optimized in future to be used in AR.

The rest of this paper is structured as follows: section 2 provides a brief introduction to dehazing and marker detection, while also presenting some current methods used to approach these problems. Section 3 defines the evaluation of dehazing methods carried out in this paper. The following section shows, explains and discusses the results measured in the evaluation. The last section concludes this paper, while providing suggestions for future work.

2. BACKGROUND

Dehazing is needed in areas where the quality of acquired images is decreased by the influence of the surrounding medium. While this medium is usually the atmosphere, and the effects affecting images are caused by weather, digital images are captured also in other environments, especially in water. Designing a successful dehazing method requires understanding of physical properties of the medium and the propagation of light in this medium. This section describes basic principles of dehazing algorithms and shows existing dehazing methods. The use of dehazing is especially important in areas where image processing sensitive to image quality is used. The second part of this section provides insight in marker detection that could be used in AR.

2.1. Dehazing

Applications using computer vision algorithms require a certain degree of input image quality to provide successful results. Some methods are particularly sensitive to artifacts that may appear in images and videos captured in uncontrolled environments such as noise, low contrast, color distortions, or lack of light in the scene. As it is not possible to ensure ideal conditions in all situations, additional processing has to be implemented to mitigate the negative weather conditions in the environment. One of such methods is dehazing, which aims to improve the visibility of images that contain haze or fog. These phenomena are usually present in outdoor scenes where different weather conditions greatly influence the quality of acquired digital images.

While traditional models describing the propagation of light in a transparent medium assume the medium is the Earth's atmosphere, in fact any transparent environment where light is capable of propagation can be a subject of study. The principles of light propagation are the same in all environments; however, the exact structure of the environment results in different effects on acquired images. Dehazing methods have to be adjusted to the type of environment where they are used in order to produce optimal results.

Dehazing methods designed for underwater environment must take into account the specific conditions of such scenes. The light in water is reflected, scattered, and attenuated as it travels to the camera, because water is constituted by particles interfering with light (Dolin and Levin, 2007). However, the character of particles in water is different than in the atmosphere. First of all, the size of particles in water is greater than the wavelengths of visible light. Also, various kinds of solid particles may be substantially present in water under some conditions.

Scattering and absorption are the main factors altering the light underwater because the majority of particles in water is bigger than the wavelength of visible light. Attenuation of light is much stronger in water compared to the atmosphere. Furthermore, the amount of absorption is dramatically increasing with the wavelength of light. So the color channels of RGB images are not affected equally, which results in the red

color channel being attenuated more than green and blue. It has been observed that red light is almost completely attenuated below the depth of ∼20 m. Green light becomes insignificant in ∼50 m (Jerlov, 1976). The remaining wavelengths are capable of reaching up to 200 m deep in the water. The rapid absorption of electromagnetic radiation by water makes the acquisition of images deep under water impossible without an artificial source of light. With a different source of light, the environment has a different impact on images as well. Artificial light has a different origin and direction than natural sunlight. At the same time, the intensity of artificial light is very high, which can increase the effect of light scattering in turbid water.

Water also contains a large variety of organic and inorganic macroscopic particles collectively called *marine snow*. The composition of marine snow is complicated and varies based on the type of water, place, and temporal factors such as season or weather. While some particles in marine snow are solid, others may be transparent. The general turbidity of water affects the physical properties of light propagation in water and contributes to the unwanted blurring, attenuation, and back light occurring in underwater images.

Most dehazing methods assume a simplified physics model of fog formation such as the one presented by Duntley et al. (1957):

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

The observed image $I(x)$ is a product of the haze free image $J(x)$ attenuated by the transmission of the scene $t(x)$ and global atmospheric light A. The *scene transmittance* $t(x)$ defines the portion of light that is transmitted by the medium without scattering it. As **Figure 1** shows, this effectively means that the color observed in a pixel is a linear combination of the real color of the object it shows and the color of the atmospheric light present in the scene.

This model illustrates that the haze-free image $J(x)$ can be obtained from the hazy image $I(x)$ provided that the transmittance map of the scene $t(x)$ and the atmospheric light A are known:

$$J(x) = \frac{I(x) - A}{t(x)} + A. \tag{2}$$

This problem, however, is ill-posed because the physical structure of the scene is complex and the transmittance and airlight are not known from a single image.

To address that, some dehazing methods are designed to require more information than just a single hazy input image. There are methods using multiple images of one scene in different conditions (Nayar and Narasimhan, 1999; Schechner et al., 2001; Narasimhan and Nayar, 2003), and additional depth information about the scene acquired by special hardware (He and Seet, 2004), or defined as a geometry of the scene (Kopf et al., 2008).

However, *single image dehazing* methods are of more practical use because they can be used on any digital image without additional information. Single image dehazing can be achieved either by *image enhancement* or *image restoration* methods. Image enhancement methods do not consider the physics model of light propagation and fog formation in removing the haze.

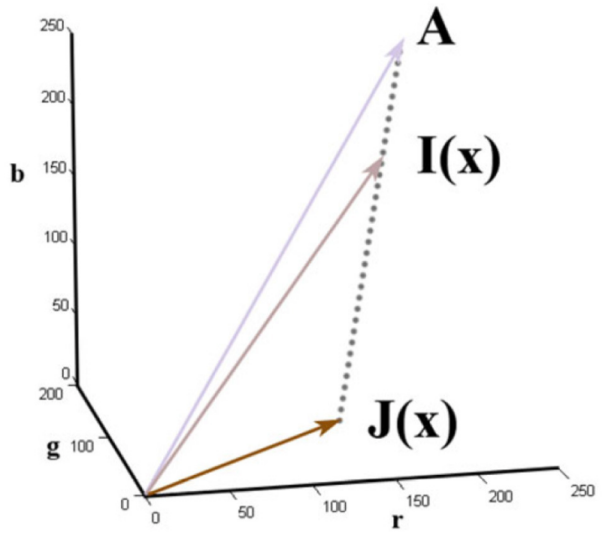


FIGURE 1 | The image formation model used in dehazing, taken from He et al. (2011).

Instead, the degraded images are processed by traditional image processing methods in order to remove noise and enhance the color, sharpness, or contrast. Image enhancement can be achieved by techniques such as the *Histogram Equalization* (Agaian and Roopaei, 2013), *Unsharp Masking* (Deng, 2011), or the *Probability-based* (Fu et al., 2015) method. One of the approaches in image enhancement techniques is to remove uneven illumination and preform color balancing. A contrast enhancement method of Morel et al. (2014) uses *Screened Poisson Equation* as a high-pass filter to remove uneven illumination while preserving details of the image, and includes a simple color balancing method to equalize colors of the result. Some enhancement methods are based on modeling mechanisms of the human visual system. One such method is the *Automatic Color Enhancement* (Getreuer, 2012) that adapts local contrast and equalizes colors similarly to the human eye. Mangeruga et al. (2018) created a detailed comparison of underwater image enhancement methods and evaluated them using quantitative metrics in different environmental conditions. Although effective in some cases, these approaches can be sensitive to noise and their results often suffer from over-enhancement.

Methods of image restoration are more complicated and usually take the depth and the physics properties of the scene into account. As the laws of light propagation in the atmosphere are similar to the laws of light propagation in water, it is possible to use some dehazing techniques targeted at removing fog or haze to restore underwater images. Fattal (2008) uses an assumption that the surface shading and the medium transmission are locally uncorrelated. They use an independent component analysis to separate the two functions and recover an image without haze from them. Although this approach proved to be effective in restoring haze, it can not handle images heavily affected by fog because the assumption may be broken. Furthermore, Fattal

presents another dehazing technique based on a principle called *color lines* (Fattal, 2014). In this method the formation of an image is explained by the color lines principle and is used to restore the haze-free image. Tan (2008) proposed an algorithm maximizing the contrast of a single hazy image by assuming that the contrast in a haze-free image is higher than in an image affected by haze. This observation is used to design a cost function in the framework of Markov random fields and the input image is restored by optimizing this framework. Li et al. (2015) perform single image dehazing using the *change of detail* prior.

He et al. (2011) proposed a *dark channel prior* (DCP) by noticing that in haze-free images, in every pixel's regional neighborhood at least one color channel contains pixels with very low intensity. The low intensity is caused either by shadows cast by objects or by very bright colors that have a high intensity in only one or two color channels. They proposed a *dark channel* of an image, which is the regional minimum of intensities across color channels in every pixel. Their dehazing technique is based on the observation that the haze increases the intensity of an image's dark channel proportionally to the amount of haze, effectively approximating the transmission of the scene. This prior has been adjusted and extended in a number of methods (Gao et al., 2014; Liu et al., 2016, 2017; Xiao et al., 2017).

There are some dehazing techniques making use of latest advancements in using neural networks. Ling et al. (2016) developed a deep transmission network to robustly restore hazy images with the use of all color channels and local patch information. A *convolutional neural network* (CNN) consisting of several convolving and max-pooling layers has been developed for image dehazing by Cai et al. (2016). Their CNN creates the transmission map of input images that can be used to remove haze by traditional techniques. Ren et al. (2016) use two convolutional neural networks to first estimate the transmission of the scene coarsely and then refine it into a precise transmission map. The training stage of techniques based on neural networks, however, is their biggest disadvantage because it requires a large amount of training images with known ground truth. As large datasets of hazy images with ground truth are not available, most of these techniques rely on synthesizing haze in datasets of images with depth maps. An RGBD dataset suitable for this was presented in Silberman et al. (2012).

General dehazing methods used for outdoor images in the atmosphere may fail to restore images acquired under water. Methods designed for underwater conditions have been created to address specifics of underwater scenes. Ancuti et al. proposed a dehazing method based on *multiscale fusion* (Ancuti et al., 2012) of two enhanced versions of a single input image. The two enhanced images derived from the input are created to improve the contrast and details in the result. Subsequently, the two images are fused together using a multi-scale fusion process guided by weight functions to create an optimal result across different scales.

The dark channel method has inspired several methods (Chiang and Chen, 2012; Wang et al., 2015; Gao et al., 2016) specialized in underwater dehazing. The method described by Chiang and Chen (2012) uses the unchanged dark channel prior to initially estimating the depth of the scene. Additional steps are implemented to handle specifics of underwater environments and to compensate the effects of artificial light that might be present in underwater images. Gao et al. (2016) adjusted the dark channel prior to underwater scenes by defining a *bright channel prior*.

Other methods make effort to use the different rate of attenuation in different wavelengths of light inside water. Galdran et al. (2015) remove the color distortion of underwater images by restoring their highly attenuated red color channel. In Carlevaris-Bianco et al. (2010), shows a simple method using the difference between intensities in color channels to estimate the depth map of an underwater image. The recovered depth is used to determine the transmission of the medium. Alternatively, Wen et al. (2013) redefined the dark channel prior only on the blue and green color channels, omitting the red channel completely.

The key step in dehazing an image is to estimate the depth of the scene in the image. A sequence of consecutive images, or a video stream can be also used to restore depth of the acquired scene. Drews et al. (2015) use the depth restored from the sequence of underwater images in a model-based dehazing approach to improve the ability to identify and match features in the images.

2.2. Marker Detection

The popularity of AR has been increasing in recent years. Its wide availability allows more applications to be developed across many areas. To determine the exact location and orientation of users and objects in AR, the objects of interest have to be detected and identified and then their orientation is computed. Different methods of tracking and detecting objects are used in AR applications. While some make active use of sensors and signals transmitted and received by the device and objects, the majority of applications rely only on visual information from a camera and the use of computer vision algorithms.

Two-dimensional markers detected in video are a common way of performing tracking for needs of AR applications because it is a cheap solution and provides good tracking results. The main disadvantage is that the environment needs to be populated with markers prior to the AR experience. Markerless-based tracking is a very popular approach nowadays but underwater is not easy to be applied because the geometry of the environment is changing its shape constantly (i.e., due to vegetation). We chose to use marker detection in our evaluation because of its simplicity. Markers provide a good means for testing in the complicated underwater environment.

Existing marker detection algorithms use different approaches to achieve better robustness, faster detection, and reliable differentiation between markers. Corners detected on a square-shaped marker are sufficient to compute the position of an object using triangulation. The inner area of square markers can be used to carry additional information making the markers distinguishable if multiple markers are used. Marker recognition using binary codes in form of a 2D matrix inside square markers has shown to be very robust, especially if it utilizes error-correcting codes (Kato and Billinghurst, 1999; Garrido-Jurado et al., 2014).

Although simply detecting at least three corners of a square marker is enough to compute its position and orientation, there are many ways to detect different types of markers providing other advantages. Markers with more complicated shapes are used to provide more accurate tracking robust to occlusions (Bencina et al., 2005; Bergamasco et al., 2016). However, it is more difficult to use the inner area of irregular shapes to carry additional information. In addition, more computational power is needed to detect the whole boundary of irregular shapes.

Little research has been done in successfully using underwater AR because of a number of constraints. Only people with sufficient training and equipment are able to dive. Diving requires substantial amount of money for logistics and adjustments to used equipment are hard to obtain. Moreover, physical conditions in water make it impossible to use some of the AR approaches such as GPS tracking. As a consequence, most of the research conducted in the area of underwater AR is tested and evaluated only in artificial conditions of a water tank. dos Santos Cesar et al. (2015) compares marker detectors in a simulated underwater environment in different levels of turbidity, lighting conditions, viewing distances, and viewing angles.

One of the areas where AR has been successfully used in real-world underwater conditions is marine archeology. AR applications developed in Bruno et al. (2016a,b) have been used in waterproof tablets at an archaeological site submerged in water. The tourists diving at the site were given tablets that guided their way and provided additional information about artifacts spotted. The application used ultra-sound sensors to track the position of the user and to register the virtual scene to the real world.

3. MATERIALS AND METHODS

This paper aims to evaluate how pre-processing by dehazing algorithms affects the success of marker detection. Numbers of successful marker detections in unprocessed underwater videos are compared to the numbers of successful marker detections in the same videos after processing it by a dehazing algorithm. The markers were also identified by the tracking, allowing to compare which marker was identified when detected. The evaluation of every test video is then based on comparing the number of successful marker detections when unprocessed, and pre-processed by each dehazing algorithm, respectively. In addition to the total counts of the successful marker detections, the comparison shows the numbers of detections that were added and that were lost in processed video compared to the unprocessed version.

The marker detection algorithm used in the evaluation is implemented in the *OpenCV 3.2.0* library. The ARUco marker detection algorithm from the ARUco AR library (Garrido-Jurado et al., 2014) provides reliable real-time marker detection robust to different lighting conditions. The detection works on gray-scale images and is based on detecting a square-shaped marker with additional information encoded in the inner area of the marker. The markers are identified by a 6×6 binary matrix encoded with error-correction that is capable of correcting up to 6 incorrectly detected bits.

Compared videos contain markers from ARUco's *DICT_6X6_50* dictionary printed on a sheet of paper and plasticized. There appears one A4 size paper with a single larger marker (∼15 cm wide) and another A4 paper with six smaller markers (∼7 cm wide) arranged in a grid. **Figure 2** shows the marker configurations. While most of the videos capture only one of the two papers, there are videos with both single and multi marker papers. Each of the seven markers is considered independently in the evaluation and the detection of each marker is evaluated for every frame of the test videos.

3.1. Compared Dehazing Methods

The evaluation compares selected state-of-the-art methods performing single image dehazing specialized for underwater environments that use different key approaches. The methods were selected for their good performance in enhancing contrast and improving visibility in a wide range of environments. The selection is based on an evaluation comparing underwater image enhancement techniques in Mangeruga et al. (2018), or the evaluations with other existing methods in the papers where they were presented (Ancuti et al., 2012; Gao et al., 2016).

3.1.1. Multi-Scale Fusion

The fusion algorithm presented in Ancuti et al. (2012) restores colors and enhances the contrast of underwater images using two different restored versions of the input image. After the two enhanced images are derived, they are used as an input to a multi-scale fusion process that combines them based on several weight functions computed from each image. To simplify the restoration process, the two input images are derived using simple image processing methods rather than complex physical properties of the scene.

The first enhanced image is obtained by a modified Gray-World (Buchsbaum, 1980) white balancing approach to correct colors. To reduce noise and improve contrast in the underwater scene, a second enhanced image is obtained from the color-corrected image created in a previous step by applying a bilateral filter followed by an adaptive histogram equalization.

A dehazed output image is created by blending the enhanced versions with a multi-scale process using image pyramids. The ratio of blending the images is determined by defining weight maps for each enhanced image based on four measures: laplacian contrast, local contrast, saliency, and exposedness.

This method was implemented in Matlab using its Image Processing Toolbox in combination with the DipImage plug-in. The saliency weight map is computed using the Matlab source code available in Achanta et al. (2009). **Figure 3** shows an image processed by this algorithm along with its two enhanced input images.

3.1.2. Bright Channel Prior (BCP)

The BCP underwater dehazing method was presented by Gao et al. (2016) and originates from a dehazing technique for atmospheric images developed in He et al. (2011). He et al.

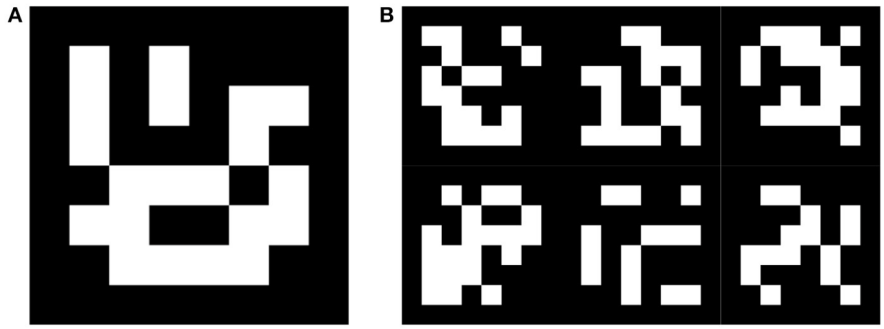


FIGURE 2 | Single marker **(A)** and a grid of multi markers **(B)** from the ARUco dictionary used in test videos.

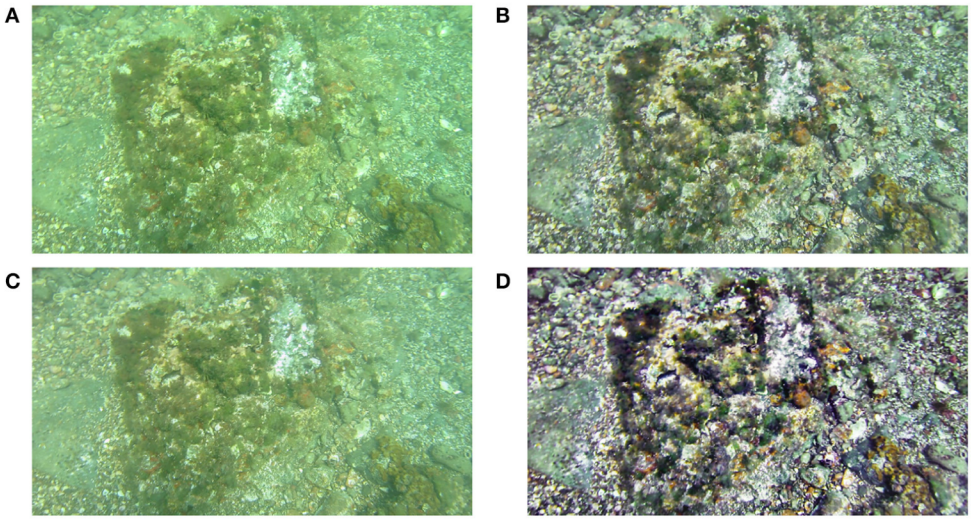


FIGURE 3 | Example of an underwater image of a seafloor **(A)** processed by the fusion algorithm **(B)**. Images **(C)** and **(D)** depict the two enhanced images created by the algorithm. The result is a fusion of the two enhanced images.

observe that one of the color channels of a local area in haze-free images has low intensity and call this assumption the Dark Channel Prior. They statistically show that this assumption is broken in hazed areas in an image. This fact can be used to estimate the amount of fog present in the image.

While this assumption works for regular scenes, it fails for images taken underwater because the dark channel of hazy areas does not differ from the dark channel of haze-free areas. The Bright Channel Prior method originates from this idea, and takes into account the properties of light absorption in the water. Visible light with longer wavelength is absorbed more quickly, which results in underwater scenes being in general green or blue due to very low intensities of the red color channel. Because of this, the BCP method considers adjusted values of color channels and defines the Bright Channel image as follows:

$$J^{bcp}(x) = \max_{y \in \Omega(x)} \left(\max_{c \in \{r,g,b\}} I^{new}(y) \right) \tag{3}$$

where $\Omega(x)$ denotes a window neighborhood centered at pixel x, and I^{new} is the adjusted input image, with the unchanged red color channel r, and inverted green and blue channels g and b respectively. It is observed that the bright channel image of the haze-free underwater images have intensities close to 1. The value of the bright channel is lower in pixels where haze is present.

The method using BCP to restore images first estimates the bright channel, using the Equation (3). To improve stability, the initial bright channel image is further rectified by blending it with a Maximum Color Difference image defined as

$$I^{mcd}(x) = 1 - \max(C_{max}(x) - C_{min}(x), C_{mid}(x) - C_{min}(x), 0) \tag{4}$$

where $C_{max}(x)$ is the color channel with the highest intensity in the pixel x, $C_{mid}(x)$ is the channel with the medium intensity, and $C_{min}(x)$ is the channel with the lowest intensity in the pixel x. The Bright Channel and the Maximum Color Difference images are blended by the proportional coefficient λ. The value of the proportional coefficient is set to be higher than 0.5 so that the Bright Channel would be the main part of the rectified result. In

practice, it is set as the maximum value of the saturation channel of the input image in HSV color space.

Another estimation needed to be done to restore the degraded image is the atmospheric light, which is a source of color distortion in the image. Among the one percent of darkest pixels in the dark channel image, the pixel with the least variance in the respective variance image is selected as the atmospheric light.

Transmittance describes the properties of the environment, which allows removing the haze from the image. It is derived from the rectified bright channel image and estimated atmospheric light by following equation

$$t^c(x) = \frac{(J^{bcp}(x) - A^c)}{1 - A^c} \tag{5}$$

where c denotes one of the three color channels, $t^c(x)$ denotes the transmittance in the respective channel, and A^c denotes the channel of the estimated atmospheric light. The average value across the different color channels in computed transmittances is set as the initial transmittance image. This result still contains some halos and artifacts introduced by the bright channel image. Therefore, the preliminary transmittance is refined by using a guide filter (He et al., 2013) with the grayscale version of the input image as the guide image.

With the transmittance image and the atmospheric light, it is possible to restore the degraded input image by the equation derived from the physics model of the image formation as shown on Equation (2). In this formula, the haze-free image $J(x)$ can be obtained from the degraded underwater image $I(x)$, estimated atmospheric light A, and the transmittance $t(x)$. Finally, the resulting image should be further corrected by the deduced histogram equalization process. The equalization should rectify the color distortion that occurs due to differences in transmittance of each color channel.

The evaluated Matlab implementation of the BCP dehazing method is using the built-in capabilities of the Image Processing Toolkit, following the equations stated above. The bright channel image is computed with the neighborhood size of 7. The same size of neighborhood is used to compute the variance value of pixels in an atmospheric light estimation step. To refine the transmittance image, the following parameters are used in Matlab's *imguidedfilter* function: *NeighborhoodSize* of 50 and *DegreeOfSmoothing* with the value of 0.01. The last step of the histogram equalization was omitted in this implementation completely. An example of an image processed by this method is in **Figure 4**.

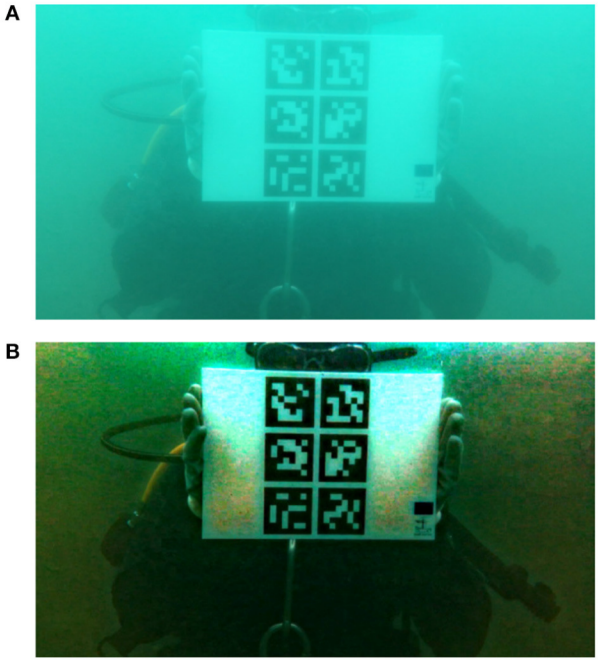


FIGURE 4 | Underwater image showing markers **(A)** processed by the BCP method **(B)**. Subject provided written informed consent for publication of this image.

3.1.3. Automatic Color Enhancement (ACE)

The ACE technique of image enhancement was first developed by Gatta et al. (2002). This method is based on a number of basic mechanisms of human visual perception. The first mechanism, *Gray World*, means that the average perceived color is gray. The *White Patch* mechanism normalizes the colors perceived toward a white reference. The mechanism of *Lateral Inhibition* causes the contrast of neighboring objects to be increased. Furthermore, this method also models the local/global adaptation mechanism of human vision.

As the original definition of this technique was computationally complex, Getreuer (2012) proposed a fast approximation of this method. The approximation reduces the complex $O(N^4)$ ACE computation into convolutions, reducing the complexity to $O(N^2 \log N)$.

The evaluated videos were processed by the ANSI C code provided by Getreuer (2012) adapted to process videos frame by frame. All test videos were processed by the level interpolation with eight levels. The weight parameter α was set to 5 and the weighing function ω was set to $1/||x-y||$. An example of an image processed by the ACE method is shown in **Figure 5**.

3.1.4. Screened Poisson Equation for Image Contrast Enhancement (SP)

The SP presented in Morel et al. (2014) represents an approach of image enhancement. It uses the *Screened Poisson* equation in the Fourier domain to remove uneven illumination in the image. The equation is used in each color channel independently and each channel is also processed by a simple color balancing algorithm (Limare et al., 2011) which equalizes the histogram of the image before and after the Screened Poisson equation solving. The method acts like a high-pass filter that in result enhances colors and the contrast in the processed image.

This is controlled by two parameters: a trade-off parameter α controlling the amount of enhancement in the screened poisson equation; and s, specifying the level of saturation in the color balance algorithm. In all processed videos the parameters were set to $\alpha = 0.0001$, and $s = 0.2$. The implementation of this method

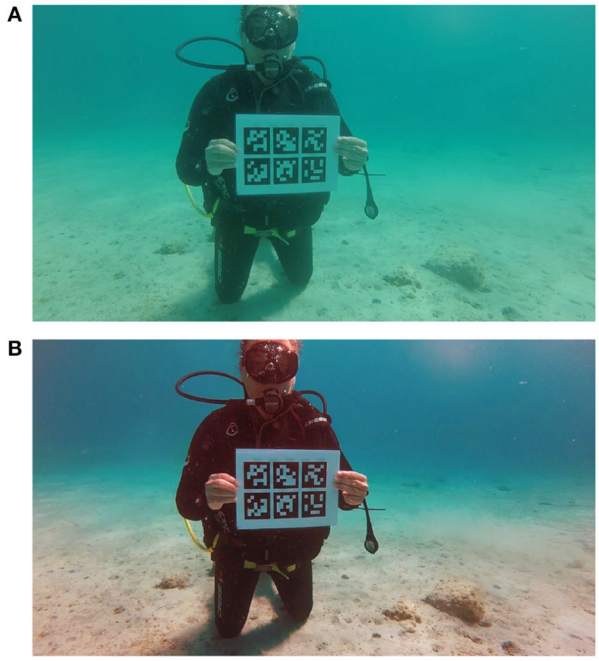


FIGURE 5 | Image with a diver holding a marker from test video 2_2 **(A)** processed by the ACE method **(B)**. Subject provided written informed consent for publication of this image.

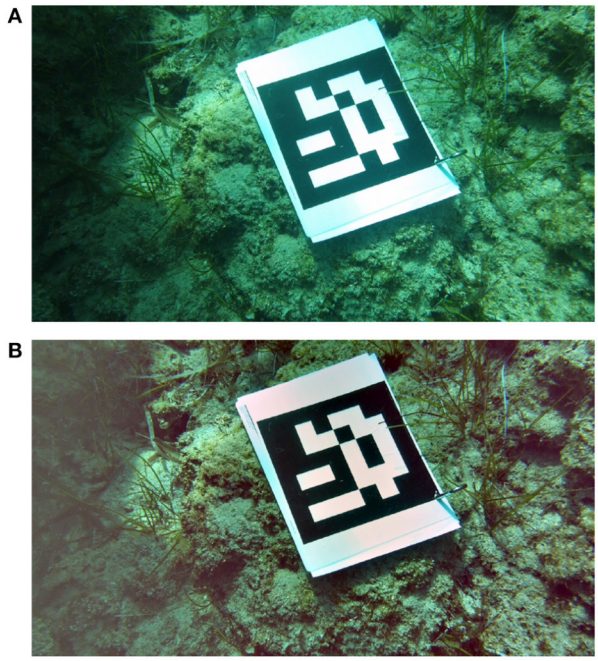


FIGURE 6 | Image of marker on a seafloor **(A)** processed by the SP algorithm **(B)**.

was adapted from the **Supplementary Materials** of Morel et al. (2014) and is written in ANSI C. An open source library FFTw[1] performs the Fast Fourier transform with required boundary conditions needed by this algorithm. A sample image processed by this method is in **Figure 6**.

3.2. Test Data

The evaluation of dehazing methods is a difficult task because the ground truth data is often not available. In addition, it is difficult to acquire visual data underwater as it requires specialized equipment, trained personnel, and other logistics. One possibility to create a dataset with ground truth images not affected by water is to set up an artificial environment in a water tank. However, the conditions simulated in an artificial environment can not imitate the full range of different real-world environments in water.

The test data set used in this research consists of videos taken in various sites in the Mediterranean Sea (Italy, Greece, and Cyprus). To make the test data set more robust, it includes data representing different kinds of environmental conditions that occur underwater. The factors affecting the conditions are the depth of the water, level of turbidity, characteristics of light in the scene, and the presence of large-scale particles.

The data used for evaluation of dehazing algorithms in this paper was taken from the underwater videos available in the *i-MareCulture project* (Skarlatos et al., 2016). There are four groups of test videos captured on different sites:

1. The first test data set was acquired in May 2017 at Nisia, Cyprus. A Nvidia tablet enclosed in a waterproof case was used to capture short uncompressed video sequences in a depth of 9 m. This is the only set with data without compression, which simulates the way the videos would be pre-processed if dehazing was used to improve an AR application in practice. The videos show single and multi marker papers in different situations including lying on the seafloor or being attached to a statue.
2. The second set of videos was shot in August 2017 near Athens, Greece. Two videos capture divers in a depth of ∼10 m holding the markers in their hands, and one video shows markers lying on the seafloor. In all videos the camera moves closer or further from the markers while focusing on them. This is the only group containing videos with both single and multi marker papers at the same time. The camera used for this group of videos is a GoPro HERO4.
3. The third set of compared videos was shot in September 2017 at the Konstantis artificial reef in Cyprus. These videos were captured in a depth of 22 m which means that the videos are strongly affected by attenuation in red and green color channels. Only the multi marker paper is used in this group of videos. The marker is positioned on the top of a metal shipwreck sunk at the bottom of the sea. This set of videos was shot on a Garmin VIRB WE waterproof camera.
4. The last group of videos used for comparison was acquired with an iPad pro 9.7 in a depth of 5–6 m. The videos show a diver holding the paper with either single or multi marker while the camera is changing the distance to the diver slowly.

[1]FFTw open source library, available opnline at http://www.fftw.org/

These videos were captured in the Underwater Archaeological Park of Baiae near Naples, Italy in October 2017.

The full technical details about each test video can be found in **Table 1** and **Figure 7** which show sample images from all groups of videos.

4. RESULTS

In this chapter, the results of the proposed evaluation of dehazing algorithms are presented. Results of each compared algorithm are presented separately and summarized in tables.

The marker detection in all evaluated videos was first inspected without any preprocessing. **Table 2** shows the total number of successful marker detections in the original videos. Although the quality of the videos is decreased by the underwater environment, the marker detection is still working if the marker is well-visible or if the marker is very close to the camera. However, the visibility conditions in the videos in group 3 were causing the marker detection to fail significantly, especially for the videos showing the single marker.

The number of successful marker detections was also inspected in test videos processed by all compared dehazing algorithms. Averaged results of the methods by the test video group are shown in **Table 3**. Full results of marker detection in the entire dataset are available in **Supplementary Material** in form of a spreadsheet. **Figure 8** shows demonstrative graphs comparing marker detection results in selected videos from all four locations. These graphs show how the dehazing methods compare to each other and to the result of marker detection with no dehazing. Each test video group is represented by a graph with the most typical results for its group.

The measures showing the improvement of marker-based AR tracking are the number of marker detections that are newly found in the processed videos, and the number of marker detections that are lost, compared to the detections in the original videos. The comparison shows that all methods can improve the marker detection by detecting markers that were not detected in the video without the preprocessing. At the same time, however, there were some markers detected in the original videos that were not detected in the videos after processing by dehazing. This happens due to over-enhancement or artifacts that a dehazing method can cause.

In general, the fusion algorithm causes the detection to lose a portion of the markers and shows improvement in the detection only in some of the videos. The relative amount of marker detections lost is much more significant than the amount of newly found detections. The percentage of detections lost after processing test videos with the fusion method reached up to 45% and at least some detections were lost in each of the test videos, except the one where no marker was detected without preprocessing. On the other hand, the amount of newly detected markers in videos processed by the fusion method is insignificant in most videos.

The results of other compared methods (BCP, ACE, SP) show similar results. The amount of newly detected markers is higher than the number of lost detections in almost all of the test videos, while the number of lost detections is not exceeding 18%. Graphs in **Figure 8** also show that these three methods yield similar numbers of successful detections, while keeping the number of successful detections higher than with no preprocessing.

The evaluation of the selected dehazing algorithms shows that dehazing can be used to improve the success of marker detection in different underwater environments. Results also suggest that the SP algorithm is improving marker detection slightly more than the other methods. However, numbers of successful marker detections show that the success of the dehazing is not the same in different underwater environments. However, the results of the methods BCP, ACE, and SP are consistently similar to each other, and show an improvement in different environments.

The first group of videos is showing considerable increases of new marker detections, as well as a high number of detections that were lost after dehazing. The results are bad especially in videos 1_4, where the number of lost detections was much higher than the newly detected markers for all compared methods. In the second group of test videos, the video 2_2 had an exceptionally high rate of detection loss. The marker detection in other videos of this group was successfully improved by all methods except for the fusion method, which failed to make a significant improvement. Except for the video 3_1, the number of newly detected markers in the third set of videos was equally low for all methods. The fusion algorithm caused much more markers to be lost after the preprocessing than the other methods in videos from this group.

The last set of test videos contains some extreme cases of results. The video 4_2 had no markers detected in its original version, so the percentage of lost and newly found markers could not be determined. The rest of the test videos in this group also had very low numbers of markers detected without preprocessing. The dehazing algorithms dramatically increased the success of marker detection in all the videos and the low

TABLE 1 | Summary of all videos used in evaluation.

Name	Format	Date	Depth (m)	Device	Resolution	Time
1_1	bytes	May	9	Nvidia tablet	1,280 × 720	00:12
1_2	bytes	May	9	Nvidia tablet	1,280 × 720	00:13
1_3	bytes	May	9	Nvidia tablet	1,280 × 720	00:14
1_4	3GP	May	9	Nvidia tablet	1,920 × 1,080	01:21
2_1	MP4	Aug	10	GoPro4	1,920 × 1,080	00:19
2_2	MP4	Aug	10	GoPro4	1,920 × 1,080	00:31
2_3	MP4	Aug	10	GoPro4	1,920 × 1,080	00:20
3_1	MP4	Sep	22	VIRB XE	1,920 × 1,440	00:15
3_2	MP4	Sep	22	VIRB XE	1,920 × 1,440	00:13
3_3	MP4	Sep	22	VIRB XE	1,920 × 1,440	00:13
3_4	MP4	Sep	22	VIRB XE	1,920 × 1,440	00:22
4_1	MPEG	Oct	10	iPad pro	1,920 × 1,080	00:59
4_2	MPEG	Oct	10	iPad pro	1,920 × 1,080	00:18
4_3	MPEG	Oct	10	iPad pro	1,920 × 1,080	01:10
4_4	MPEG	Oct	10	iPad pro	1,920 × 1,080	00:14

The names of videos show the number of test videos set and its number in the set.

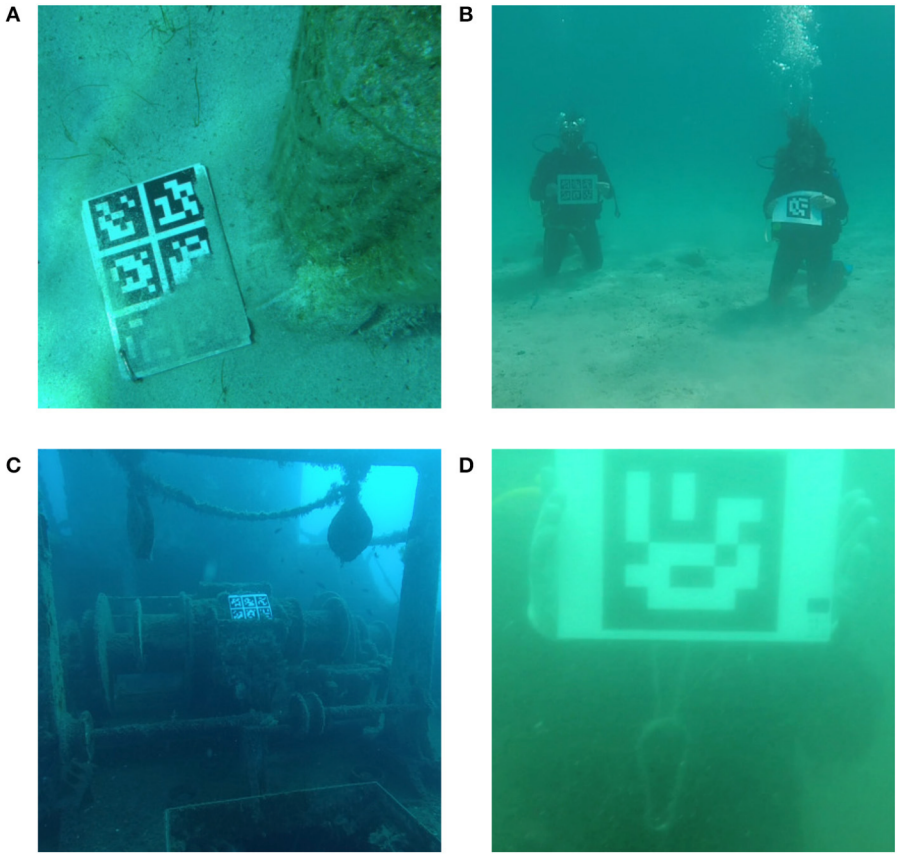


FIGURE 7 | Sample image from test videos in group one **(A)**, two **(B)**, three **(C)**, and four **(D)**. Subjects provided written informed consent for publication of this image.

TABLE 2 | Results of marker detection in evaluated videos without preprocessing.

Name	Single marker	Multi marker	Total
1_1	0	343	343
1_2	63	0	63
1_3	117	0	117
1_4	0	8,190	8,190
2_1	560	3,349	3,909
2_2	527	4,811	5,338
2_3	588	0	588
3_1	247	0	247
3_2	1,127	0	1,127
3_3	863	0	863
3_4	930	0	930
4_1	4	0	4
4_2	0	0	0
4_3	0	436	436
4_4	0	72	72

TABLE 3 | Averaged results of the numbers of successful marker detections in whole test video groups by dehazing method.

Test videos group	Method	Average detections lost %	Average detections added %
Group 1	fusion	16.90	09.58
Group 1	bcp	10.60	16.59
Group 1	ace	10.33	09.79
Group 1	sp	10.37	10.86
Group 2	fusion	02.30	01.38
Group 2	bcp	00.29	03.82
Group 2	ace	00.24	04.22
Group 2	sp	00.25	04.32
Group 3	fusion	22.19	01.66
Group 3	bcp	04.07	06.07
Group 3	ace	03.77	09.19
Group 3	sp	08.81	08.99
Group 4	fusion	62.60	117.59
Group 4	bcp	00.00	7099.22
Group 4	ace	00.00	4633.30
Group 4	sp	00.00	8967.52

amount of successful marker detections in the original means that the percentage of lost marker detections can be very high but insignificant to the evaluation. The BCP, ACE, and SP methods were successful in detecting markers in all four test videos in this group while not losing any markers. In contrast, multi-scale fusion dehazing improved the marker detection only in two out of four videos.

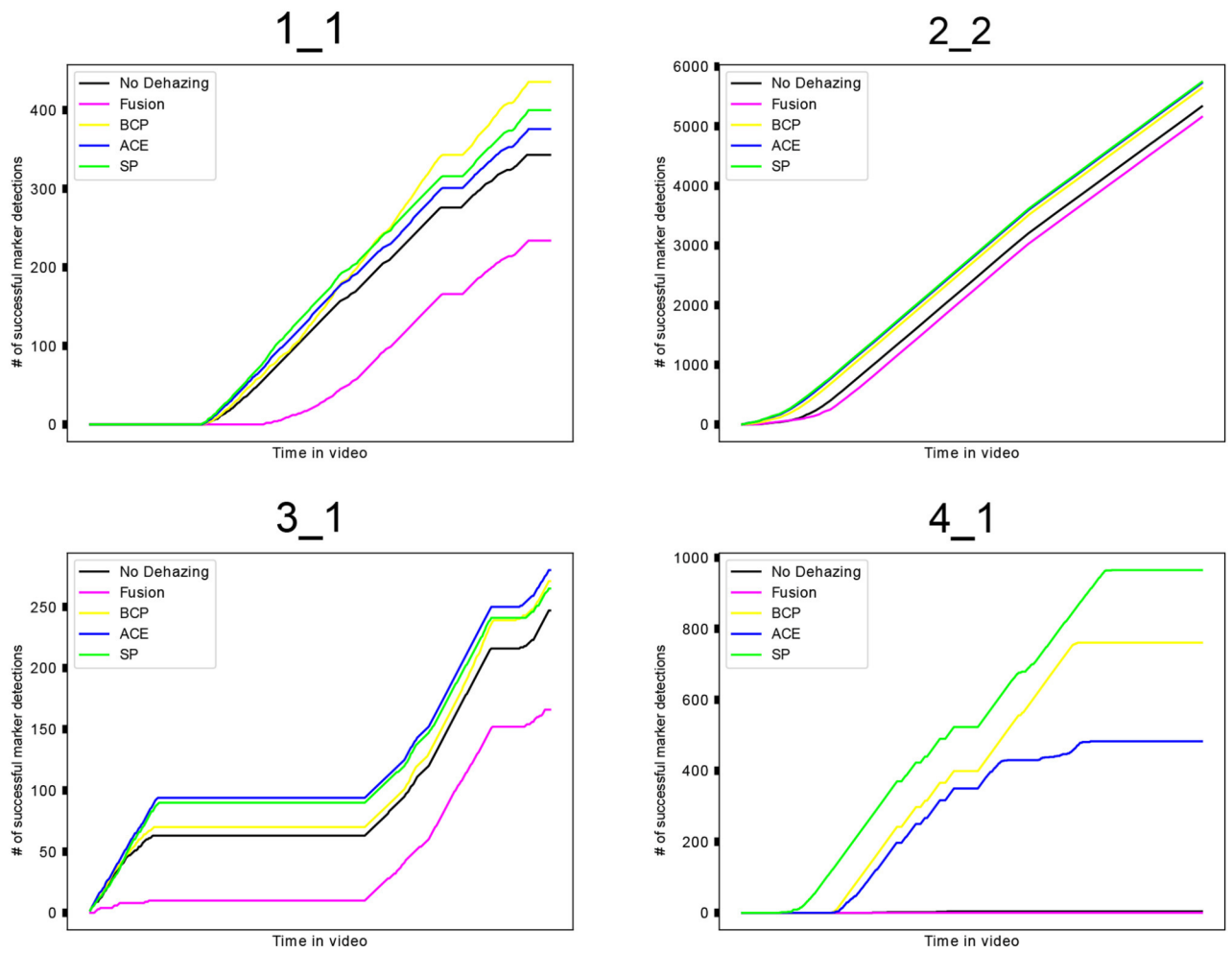


FIGURE 8 | Results of marker detection shown on graphs. Each test video group is represented by one video (labeled on top). Graphs are showing the cumulative number of successful marker detections in videos for each dehazing method or no dehazing, respectively.

Although the dehazing algorithm evaluation was aiming to cover a wide range of possible inputs, the variability of visibility conditions underwater is much more diverse. The evaluation could be improved by using larger a amount of data from more different sites. More focus in the evaluation can be put on considering the specifics of location and current condition in each test video to understand more how these factors affect the outcome of dehazing methods.

As the dehazing algorithms are usually not designed to work in real-time, their performance is often time-consuming and they are not designed to be fast. However using dehazing to improve tracking in an AR application will require the dehazing method to be able to run in real-time performance. Therefore, the evaluation also considers the speed of compared algorithms. The fusion algorithm was the slowest one, having an average processing time of 7 s per frame. The BCP algorithm consists of simpler steps that would allow for optimizations. The unoptimized implementation used in the comparison took an average three seconds to process a single frame of the video. The implementations of ACE and SP were faster, both needing ∼2 s to process a single image. All times were measured on images with a resolution of 1,920 × 1,080 pixels on an HP laptop computer with 2.40 GHz Intel i7-5500U CPU and 8GB of RAM.

At the moment, the processing time of the compared methods is not suitable for a real-time application. However, this evaluation shows the capabilities of dehazing in improving marker detection and it can be used in the future to develop an optimized method running in real-time.

5. CONCLUSIONS

This paper presented current methods of underwater image dehazing that restore the visibility decreased by underwater environments. Selected underwater dehazing methods were evaluated to show their capability of improving marker-based tracking for augmented reality applications.

The evaluation of dehazing techniques was carried out by comparing the number of successful marker detections in a number of test videos. The existing marker detection methods were first described and the ARUco library was used to detect

markers in the evaluation. For evaluation, four dehazing methods targeted for underwater scenes were selected from existing state-of-the-art techniques and implemented. The multi-scale fusion algorithm restores the images by creating two different enhanced versions of an input image and combines them in a multi-scale fusion process according to weights measuring the quality of enhancement in the two derived images. The bright channel prior algorithm defines a special bright channel of the image and uses it to estimate the transmittance of the scene. The transmittance is then used to restore the image according to the formula of image formation. The automatic color enhancement and screened poisson methods represent methods of image enhancement and increase the contrast of the underwater images.

The results of the dehazing algorithm evaluation suggested that the SP algorithm performs the best in improving the marker detection. The BCP and ACE methods also showed significant improvement in marker detection, although not as good as SP.

Furthermore, a different level of success in improving the tracking was noticed in different groups of videos. This shows that the marker tracking performance may differ greatly according to depth, location, and the actual light and turbidity conditions. In addition, it has been shown that in some cases the dehazing caused a loss of some markers that were detected in the video without preprocessing.

In the future, there are many opportunities to improve and extend the evaluation presented in this paper. Extending the evaluation with a larger and more diverse data set would be beneficial. In addition, other dehazing methods should be evaluated as there are many recent approaches addressing the problem of underwater dehazing. The results of the evaluation could be also supported by using data with ground truth acquired in controlled conditions.

Other measures could be also included in the evaluation to provide more insight in how dehazing affects the AR application performance. With ground truth available in the test data set, the accuracy of marker detection could be evaluated. Moreover, different parameters of the dehazing methods and AR tracking algorithms can be used to evaluate their influence on the problem.

As the dehazing algorithms proved to be useful in improving the marker detection underwater, they can be used in AR applications targeted in underwater environments. Some dehazing algorithms will allow optimization that will enable an AR application to use dehazing as a preprocessing step to effectively improve the success of tracking.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this manuscript will be made available by the authors, without undue reservation, to any qualified researcher.

AUTHOR CONTRIBUTIONS

MŽ, JČ, and FL conceived the research and defined the evaluation method. FB, DS, and FL acquired and provided test data. JČ implemented a tool for marker detection performance evaluation in videos. MŽ implemented the compared methods, carried out the processing and comparison of dehazing algorithms in test videos. MŽ wrote the paper. FB, DS, JČ, and FL provided comments.

FUNDING

This research was part of the i-MareCulture project (Advanced VR, iMmersive Serious Games and Augmented REality as Tools to Raise Awareness and Access to European Underwater CULTURal heritagE, Digital Heritage) that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 727153.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/frobt.2018.00092/full#supplementary-material

Attached will be a spreadsheet containing full results of marker detection in all test videos for all compared methods.

REFERENCES

Achanta, R., Hemami, S., Estrada, F., and Ssstrunk, S. (2009). "Frequency-tuned salient region detection," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2009)* (Miami, FL), 1597–1604.

Agaian, S., and Roopaei, M. (2013). "New haze removal scheme and novel measure of enhancement," in *2013 IEEE International Conference on Cybernetics (CYBCO)* (Lausanne), 219–224.

Ancuti, C., Ancuti, C. O., Haber, T., and Bekaert, P. (2012). "Enhancing underwater images and videos by fusion," in *2012 IEEE Conference on Computer Vision and Pattern Recognition* (Providence, RI), 81–88.

Bencina, R., Kaltenbrunner, M., and Jorda, S. (2005). "Improved topological fiducial tracking in the reactivision system," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops* (San Diego, CA), 99–99.

Bergamasco, F., Albarelli, A., Cosmo, L., Rodol, E., and se llo, A. T. (2016). An accurate and robust artificial marker based on cyclic codes. *IEEE Trans. Pattern Anal. Mach. Intell.* 38, 2359–2373. doi: 10.1109/TPAMI.2016.2519024

Bruno, F., Lagudi, A., Barbieri, L., Muzzupappa, M., Ritacco, G., Cozza, A., et al. (2016a). *Virtual and Augmented Reality Tools to Improve the Exploitation of Underwater Archaeological Sites by Diver and Non-diver Tourists*. Cham: Springer International Publishing.

Bruno, F., Lagudi, A., Muzzupappa, M., Lupia, M., Cario, G., Barbieri, L., et al. (2016b). Project VISAS: virtual and augmented exploitation of submerged archaeological sitesoverview and first results. *Mar. Technol. Soc. J.* 50, 119–129. doi: 10.4031/MTSJ.50.4.4

Buchsbaum, G. (1980). A spatial processor model for object colour perception. *J. Franklin Inst.* 310, 1–26.

Cai, B., Xu, X., Jia, K., Qing, C., and Tao, D. (2016). Dehazenet: an end-to-end system for single image haze removal. *IEEE Trans. Image Process.* 25, 5187–5198. doi: 10.1109/TIP.2016.2598681

Carlevaris-Bianco, N., Mohan, A., and Eustice, R. M. (2010). "Initial results in underwater single image dehazing," in *OCEANS 2010 MTS/IEEE* (Seattle, WA) 1–8.

Chiang, J. Y., and Chen, Y. C. (2012). Underwater image enhancement by wavelength compensation and dehazing. *IEEE Trans. Image Process.* 21, 1756–1769. doi: 10.1109/TIP.2011.2179666

Deng, G. (2011). A generalized unsharp masking algorithm. *IEEE Trans. Image Process.* 20, 1249–1261. doi: 10.1109/TIP.2010.2092441

Dolin, L. S., and Levin, I. M. (2007). "Underwater optics," in *The Optics Encyclopedia* (American Cancer Society). doi: 10.1002/9783527600441.oe099

dos Santos Cesar, D. B., Gaudig, C., Fritsche, M., dos Reis, M. A., and Kirchner, F. (2015). "An evaluation of artificial fiducial markers in underwater environments," in *OCEANS 2015 - Genova* (Genova), 1–6.

Drews, P., Nascimento, E. R., Campos, M. F. M., and Elfes, A. (2015). "Automatic restoration of underwater monocular sequences of images," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Hamburg), 1058–1064.

Duntley, S. Q., Boileau, A. R., and Preisendorfer, R. W. (1957). Image transmission by the troposphere i*. *J. Opt. Soc. Am.* 47:499.

Fattal, R. (2008). Single image dehazing. *ACM Trans. Graph.* 27, 72:1–72:9. doi: 10.1145/1360612.1360671

Fattal, R. (2014). Dehazing using color-lines. *ACM Trans. Graph.* 34, 13:1–13:14. doi: 10.1145/2651362

Fu, X., Liao, Y., Zeng, D., Huang, Y., Zhang, X. P., and Ding, X. (2015). A probabilistic method for image enhancement with simultaneous illumination and reflectance estimation. *IEEE Trans. Image Process.* 24, 4965–4977. doi: 10.1109/TIP.2015.2474701

Galdran, A., Pardo, D., Picón, A., and Alvarez-Gila, A. (2015). Automatic red-channel underwater image restoration. *J. Vis. Commun. Image Represent.* 26(Suppl. C), 132–145. doi: 10.1016/j.jvcir.2014.11.006

Gao, R., Wang, Y., Liu, M., and Fan, X. (2014). Fast algorithm for dark channel prior. *Electron. Lett.* 50, 1826–1828. doi: 10.1049/el.2014.2884

Gao, Y., Li, H., and Wen, S. (2016). Restoration and enhancement of underwater images based on bright channel prior. *Math. Probl. Eng.* 2016:3141478. doi: 10.1155/2016/3141478

Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F., and Marín-Jiménez, M. (2014). Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recogn.* 47, 2280–2292. doi: 10.1016/j.patcog.2014.01.005

Gatta, C., Rizzi, A., and Marini, D. (2002). "ACE: an automatic color equalization algorithm," in *Conference on Colour in Graphics, Imaging, and Vision* (Poitiers), 316–320.

Getreuer, P. (2012). Automatic color enhancement (ACE) and its fast implementation. *Image Process. On Line* 2, 266–277. doi: 10.5201/ipol.2012.g-ace

He, D.-M., and Seet, G. G. (2004). Divergent-beam lidar imaging in turbid water. *Opt. Lasers Eng.* 41, 217–231. doi: 10.1016/S0143-8166(02)00138-0

He, K., Sun, J., and Tang, X. (2011). Single image haze removal using dark channel prior. *IEEE Trans. Pattern Anal. Mach. Intell.* 33, 2341–2353. doi: 10.1109/TPAMI.2010.168

He, K., Sun, J., and Tang, X. (2013). Guided image filtering. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 1397–1409. doi: 10.1109/TPAMI.2012.213

Jerlov, N. G. (1976). *Marine Optics, Vol. 14*. Amsterdam: Elsevier.

Kato, H., and Billinghurst, M. (1999). "Marker tracking and hmd calibration for a video-based augmented reality conferencing system," in *2nd IEEE and ACM International Workshop on Augmented Reality* (San Francisco, CA), 85–94.

Kopf, J., Neubert, B., Chen, B., Cohen, M., Cohen-Or, D., Deussen, O., et al. (2008). "Deep photo: model-based photograph enhancement and viewing," in *ACM SIGGRAPH Asia 2008 Papers* (New York, NY), 1–116.

Li, J., Zhang, H., Yuan, D., and Sun, M. (2015). Single image dehazing using the change of detail prior. *Neurocomputing* 156(Suppl. C), 1–11. doi: 10.1016/j.neucom.2015.01.026

Limare, N., Lisani, J.-L., Morel, J.-M., Petro, A. B., and Sbert, C. (2011). Simplest color balance. *Image Process. On Line* 1, 297–315. doi: 10.5201/ipol.2011.llmps-scb

Ling, Z., Fan, G., Wang, Y., and Lu, X. (2016). "Learning deep transmission network for single image dehazing," in *2016 IEEE International Conference on Image Processing (ICIP)*, 2296–2300.

Liu, X., Zhang, H., Tang, Y. Y., and Du, J. X. (2016). Scene-adaptive single image dehazing via opening dark channel model. *IET Image Process.* 10, 877–884. doi: 10.1049/iet-ipr.2016.0138

Liu, Y., Li, H., and Wang, M. (2017). Single image dehazing via large sky region segmentation and multiscale opening dark channel model. *IEEE Access* 5, 8890–8903. doi: 10.1109/ACCESS.2017.2710305

Ludvigsen, M., Sortland, B., Johnsen, G., and Singh, H. (2007). Applications of geo-referenced underwater photo mosaics in marine biology and archaeology. *Oceanography* 20, 140–149. doi: 10.5670/oceanog.2007.14

Mangeruga, M., Cozza, M., and Bruno, F. (2018). Evaluation of underwater image enhancement algorithms under different environmental conditions. *J. Mar. Sci. Eng.* 6:10. doi: 10.3390/jmse6010010

Morel, J.-M., Petro, A.-B., and Sbert, C. (2014). Screened poisson equation for image contrast enhancement. *Image Process. On Line* 4, 16–29. doi: 10.5201/ipol.2014.84

Narasimhan, S. G., and Nayar, S. K. (2003). Contrast restoration of weather degraded images. *IEEE Trans. Pattern Anal. Mach. Intell.* 25, 713–724. doi: 10.1109/TPAMI.2003.1201821

Nayar, S. K., and Narasimhan, S. G. (1999). "Vision in bad weather," in *Proceedings of the Seventh IEEE International Conference on Computer Vision* (Kerkyra), 820–827

Ren, W., Liu, S., Zhang, H., Pan, J., Cao, X., and Yang, M.-H. (2016). *Single Image Dehazing via Multi-scale Convolutional Neural Networks*. Cham: Springer International Publishing.

Schechner, Y. Y., Narasimhan, S. G., and Nayar, S. K. (2001). "Instant dehazing of images using polarization," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Kauai), 325–332

Silberman, N., Hoiem, D., Kohli, P., and Fergus, R. (2012). *Indoor Segmentation and Support Inference from RGBD Images*. Berlin; Heidelberg: Springer Berlin Heidelberg.

Skarlatos, D., Agrafiotis, P., Balogh, T., Bruno, F., Castro, F., Petriaggi, B. D., et al. (2016). *Project iMARECULTURE: Advanced VR, iMmersive Serious Games and Augmented REality as Tools to Raise Awareness and Access to European Underwater CULTURal heritagE*. Cham: Springer International Publishing.

Tan, R. T. (2008). "Visibility in bad weather from a single image," in *2008 IEEE Conference on Computer Vision and Pattern Recognition* (Anchorage, AK), 1–8.

Wang, J.-B., He, N., Zhang, L.-L., and Lu, K. (2015). Single image dehazing with a physical model and dark channel prior. *Neurocomputing* 149(Pt B), 718–728. doi: 10.1016/j.neucom.2014.08.005

Wen, H., Tian, Y., Huang, T., and Gao, W. (2013). "Single underwater image enhancement with a new optical model," in *2013 IEEE International Symposium on Circuits and Systems (ISCAS2013)* (Beijing), 753–756.

Xiao, J., Zhu, L., Zhang, Y., Liu, E., and Lei, J. (2017). Scene-aware image dehazing based on sky-segmented dark channel prior. *IET Image Process.* 11, 1163–1171. doi: 10.1049/iet-ipr.2017.0058

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

The reviewer, AV, and handling Editor declared their shared affiliation.

Copyright © 2018 Žuži, Čejka, Bruno, Skarlatos and Liarokapis. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

EUROGRAPHICS Workshop on Graphics and Cultural Heritage (2018)
R. Sablatnig and M. Wimmer (Editors)

Improving Marker-Based Tracking for Augmented Reality in Underwater Environments

Jan Čejka[1], Marek Žuži[1], Panagiotis Agrafiotis[2], Dimitrios Skarlatos[2], Fabio Bruno[3], and Fotis Liarokapis[1]

[1] Human Computer Interaction Laboratory, Faculty of Informatics, Masaryk University, Czech Republic
[2] Photogrammetric Vision Laboratory, Department of Civil Engineering and Geomatics, Cyprus University of Technology, Cyprus
[3] 3D Research s.r.l. – University of Calabria, Italy

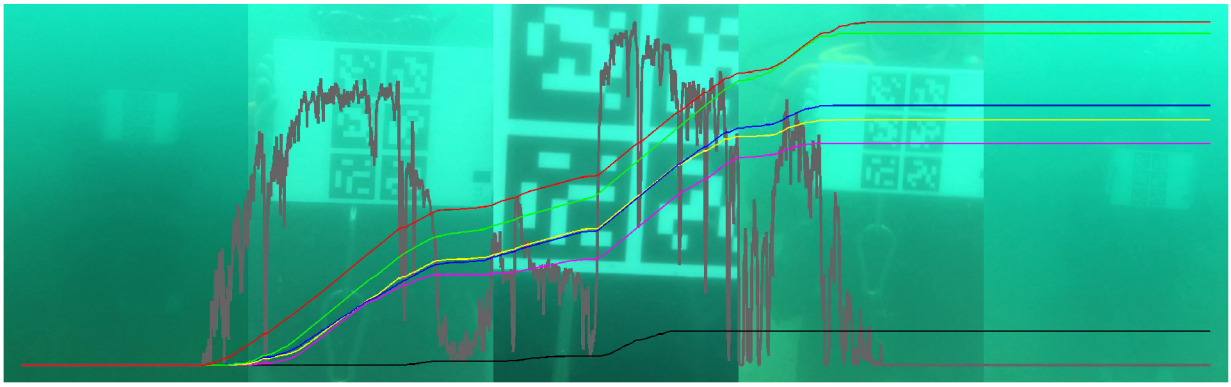


Figure 1: *Number of markers detected by ARUco in video without using any image enhancement algorithm (black), and in videos enhanced by CLAHE (magenta), debluring (yellow), white balancing (blue), and our MBUWWB algorithm (green), at different marker visibility levels (gray). Combination of fast ARUco and our fast algorithm provides results comparable with more robust and much slower AprilTag2 (red).*

Abstract

Underwater sites are a challenging environment for augmented reality. Images taken under water are degraded in several ways, most importantly they are affected by unbalanced colors due to uneven absorption of light in each color channel, blurring and desaturation caused by turbidity, or noise due to the presence of larger organisms, plants, or bubbles. In this paper, we introduce a new method based on white balancing that enhances underwater images to improve the results of detection of markers. We compare this method with several image enhancement methods, and evaluate their performance when applied to the problem of detecting markers under water. Our results show that our method improves the detection in underwater environments while keeping the computation time low.

CCS Concepts

•***Computing methodologies*** → *Mixed / augmented reality; Computer vision;*

1. Introduction

Underwater cultural heritage assets are widely spread into the Mediterranean. Unlike land archaeological sites, however, submerged settlements, ancient ports and other coastal industrial installations, especially shipwrecks, are not accessible to the general public nor all experts, due to their environment and depth. On the other hand, underwater archaeological parks already exist in Mediterranean allowing divers to explore ancient cities. Augmented reality (AR) gives us an opportunity to enhance people's vision of the real world and is the perfect medium for exploiting cultural heritage sites. One of the main technical requirements of achieving effective AR applications is accurate tracking. Although there has been a lot of research in the past few years in vision tracking, it is still almost unexplored in underwater environments and in particular in the open sea.

© 2018 The Author(s)
Eurographics Proceedings © 2018 The Eurographics Association.

The first time that underwater AR was proposed was in 1999 by Gallagher et al. [Gal99] where the development of a HMD for Navy divers was presented. In 2009, Morales et al. [MKMK09] presented an experimental AR system that provided visual aids to increase commercial divers' capability to detect, perceive, and understand elements in underwater controlled environments. In the same year, researchers developed the AREEF system which allowed people to discover underwater word of corals, fish or fairy-tale wrecks in a swimming pool in a comfortable and safe way [BBM09]. In 2013, the AREEF system was improved in order to be used by more than one person [OBLS13] and also by children [OBS16]. Also in 2013, Brown et al. [BW13] presented a system for using augmented reality under water to help divers with navigation and identification of fish species.

Conditions in sea are different, however. The vision is degraded by several factors, most importantly by turbidity, different attenuation of light in different wave lengths, and a presence of large particles in water. This has a huge impact on a detection and recognition of objects, which is an inherent part of each AR application. To our knowledge, there was only one attempt to adopt AR technology directly in the submerged environment by using underwater tablet. This device was equipped with an underwater positioning and orientation system that guided the diver tourists during the diving session while providing information about the archaeological artifacts spotted during the visit [BLB*16,BLM*16]. However, this solution was based on acoustic sensors and not vision.

The focus of this research is on improving videos taken in sea environments before they are processed by marker detecting algorithms to increase the performance of these algorithms at such places. Although similar tests were already done in laboratory conditions [dSCGF*15], as a part of an evaluation of a single image improving algorithm [AC14, GLW16], or in underwater photogrammetry [SPOP16, ADGS17], to our best knowledge there is no evaluation of real-time image enhancing algorithms focusing on a detection of markers located in underwater marine environments that can be used for AR.

This paper is based on work of Žuži et al. [ŽČB*18], which evaluates off-line algorithms for dehazing. In contrast to it, this paper focuses on real-time image enhancing algorithms. Our main contribution is a new method for real-time improvement of underwater images containing markers for AR, and comparison of its performance with three state-of-the-art methods in four different underwater open sea environments. This comparison is done based on the number of markers detected in improved videos. We demonstrate that applying online image enhancing algorithms improves the performance of marker-based tracking. Our results exhibit a clear improvement of marker-based tracking, which indicates that it can be used in the future for underwater vision tracking.

The rest of the paper is structured as follows. Section 2 presents related work done in improving images and marker-based tracking. Section 3 illustrates the proposed methodology, Section 4 describes the image enhancement algorithms that were tested, and Section 5 shows our method to estimate marker visibility. Section 6 demonstrates our results obtained from tested videos, Section 7 results of a cultural heritage use case, and Section 8 discussions. Finally, Section 9 presents conclusions and future work.

2. Related work

This section presents related work and consists of three parts: improvement of images taken in standard conditions, improvement of images taken under water, and marker-based tracking.

2.1. Improvement of images taken in standard conditions

Improvement of images is one of the most common problems solved when preprocessing images. The most important problems that are solved are removing effects of noise and sharpening of the image, while preserving colors, edges, and overall perception of objects in the image. The most common techniques used for general image improvements are median and bilateral filters [Wei06, Tsi16], and unsharp filters [KCFK17]. Lee and Woo [LW09] present white-balancing technique that improves colors of an image using data from a detected marker, but unlike us, they focus on improving image appearance and not on improving marker detection. Images taken outside of water are often degraded similarly as images taken under water. There are several techniques focusing on improving images damaged by fog or haze, some of these techniques acquire additional data about the scene to ease dehazing. Treibitz et al. [TS09] discusses advantages and disadvantages of using polarization filters to obtain several images of the same scene with different polarization of the light to suppress the effect of fog in final image. Kopf et al. [KNC*08] uses depth information at each pixel to relight the scene or remove the effect of haze.

Moreover, the problem of removing haze from single images without any other additional information available is also becoming more important topic of research. He et al. [HST11] discovers that most regions of haze-free images have at least one color channel (red, green, or blue) that contains very low values. They call this channel a *dark channel*, and proposes a *dark channel prior*, which is a method for dehazing images based on this information. Zhu et al. [ZMS15] describes a similar method that is based on a difference between saturation and value channels of pixels represented in HSV color space. In 2008, Fattal [Fat08] assumes the illumination of objects in the scene and depth of these objects are two statistically uncorrelated functions, uses methods for independent component analysis to separate these functions, and recovers the image without haze. In 2014, Fattal [Fat14] describes another technique that is based on *color lines*, a method for representing colors described by Omer et al. [OW04]. This technique processes small patches in input images with colors that belong to the same color line, and uses properties of this color line to estimate the amount of haze in this patch.

In addition to this, Zhang et al. [ZH14] describes a method that tests several possible solutions, and chooses the best solution for each pixel of input image individually. He also provides an implementation for GPUs that runs real-time. Li et al. [LTT*15] builds on methods that compute the depth of image from a video sequence, and develops a method that simultaneously reconstructs the scene and removes fog in the scene, using the depth cues from scene reconstruction for removing the fog and vice versa. Ancuti et al. [AC14] focuses especially at a detection of SIFT features for matching two images, and proposes a method that enhances contrast in images to recover and match as many features as possible.

© 2018 The Author(s)
Eurographics Proceedings © 2018 The Eurographics Association.

In recent years, there has been a lot of attention in using neural networks. Cai et al. [CXJ*16] was inspired by techniques based on dark channel, enhancing contrast, attenuation of colors, and hue disparity, and designed a convolutional neural network (CNN) that consists of several convolving and max-pooling layers to obtain a transmission map for further removal of fog. Ren et al. [RLZ*16] designs a CNN to obtain firstly an assumption of coarse transmission map, and than use this information as an input to another CNN to obtain a precise transmission map. Both these CNN do not depend on any assumptions or prior knowledge of input images, however, they need a substantial amount of training images with known ground truth to train them. This problem is tackled by Li et al. [LGG18], who proposes a neural network for improving underwater images without needing training images with known ground truth.

2.2. Improvement of underwater images

Restoration of underwater images represents a greater challenge than restoration of images taken outside of water. This is due to the fact that underwater images are degraded by much more phenomena than other images, most importantly by turbidity, floating particles, and an uneven absorption of different color channels of light. Ancuti et al. [AAHB12] proposes a system based on a fusion of multiple images that are derived from the input image using common image improving techniques like white-balancing and contrast enhancement. Li et al. [LLZ*14] creates a pipeline of several filters to improve images degraded by uneven illumination, noise, turbidity, and uneven absorption of colors. Chiang et al. [CC12] uses the dark channel prior proposed by He et al. [HST11] to estimate the depth of objects in the input image. This depth is then used to reduce uneven illumination caused by the scattering of artificial light, and to reduce the effect of turbidity in the image. Although the dark channel prior provides very good results in images taken outside of water, it cannot be used for images taken in large depths under water, due to the missing red and green channels.

Some algorithms were designed especially to overcome this issue. Carlevaris-Bianco et al. [CBME10] uses a difference between the maximum in red, green and blue channels to get a coarse initial estimate of the depth in the image. This initial estimate is refined by using a natural image matting, similarly as in a work by He et al. [HST11]. Drews et al. [DNBC16] suggests ignoring the red channel at all, and use only green and blue channels to estimate the depth in the image. Gao et al. [GLW16] presents a bright channel prior, which operates with the original red channel and inverted green and blue channels. These methods try to estimate the depth from the input image. However, the depth can be obtained using other methods. Drews et al. [DNCE15] focuses on processing a video instead of using individual images, and reconstructs the scene from it using methods for scene reconstruction. Similarly to other methods, this depth is used to reduce the effects of turbidity in images. Babaee et al. [BN15] proposes using a sonar in environments with high turbidity to obtain 3D positions of objects in the image. Cho et al. [CSK16] uses a depth precisely measured in a few points in the input image, and uses incremental Gaussian process to estimate the depth in the rest of the image.

2.3. Marker-based tracking

There are different algorithms for marker-based tracking. They use different types of markers to achieve faster detection and recognition of markers, better robustness in uneven lighting conditions, or better differentiation between individual markers when multiple markers are used. Square markers [KB99, GJnSMCMJ14, WO16, Fia05] are well detectable, and their corners are sufficient to compute marker's position. The inner part of markers is used to carry an information to distinguish between different markers. ARToolKit [KB99] uses arbitrary images, making them very versatile and user-friendly. Binary codes are also used, often in a form of a two-dimensional matrix. This form is more robust, especially when it is augmented with Hamming error-correcting codes [KB99, WO16, Fia05] or specifically created dictionaries [GJnSMCMJ14].

Markers in shapes of disks and ellipses are also used [KPS10, NF02]. Instead of using only four points to compute the position of the marker, as it is done in case of square markers, the whole boundary is used to compute the position, making it more precise and more robust to occlusions. On the other hand, spherical shape imposes difficulties when designing robust patters inside the marker. Irregular shapes of markers are also used [BKJ05, BAC*16], allowing to design very precise markers, but usually at the price of increased detection time. Performance of marker-based detectors is usually evaluated in clean view, focusing only on the size and orientation of the marker, occlusions, optics of the camera, and illumination of the marker [TATM13, GJnSMCMJ14, NF02]. Cesar et al. [dSCGF*15] compares marker detectors in a simulated underwater environment in different levels of turbidity, lighting conditions, viewing distances, and viewing angles.

3. Methodology

In this paper, two marker detection libraries were used to detect markers in our videos, ARUco [GJnSMCMJ14] and AprilTag2 [WO16]. These libraries were chosen for several reasons: their source code is available as open source and is frequently updated, they detect markers in real-time, and they are robust to different lighting conditions. We refer to corresponding papers for detailed information about these algorithms. Detectors detects markers in gray-scale images so all images are converted to YUV color space, and then markers are detected in Y channel. This color space was chosen, because it is supported by most mobile devices. An implementation of ARUco that is a part of OpenCV 3.2.0 was employed. For AprilTag2, we used source code available at the site of the project, version 0.9.8. Both libraries are optimized to use multiple threads; additionally, OpenCV is optimized to use SIMD instructions.

Performance of these two marker detection libraries was evaluated in laboratory conditions in a work of Cesar et al. [dSCGF*15]. Their results showed that AprilTag (the predecessor of AprilTag2) provided better results than ARUco in highly turbid water environments, and only a slightly better results in other environments. However, the detection time of AprilTag was approximately three times higher than that of ARUco, which can make the it unusable in real-time AR applications. We used 6 markers from ARUco

© 2018 The Author(s)
Eurographics Proceedings © 2018 The Eurographics Association.

DICT_6X6_50 dictionary. To distinguish between markers, this dictionary uses labels with 36 bits of information in a form of a binary matrix of 6 rows and 6 columns, and can correct up to 6 incorrectly detected bits. The markers were printed on a A4 paper, formed in a grid of two rows and three columns. Each marker measured 8 centimeters, and there was 1 centimeter space between each row and column. The detection of markers was evaluated on four set of videos from [SAB*16], see Figure 2. These videos were taken in Mediterranean sea; the locations were chosen by experts in underwater archeology to represent the most typical environments of archaeological sites. These videos were processed off-line on a standard PC using different image enhancing algorithms. Also, the detection of markers was performed off-line on each frame of these processed videos.

4. Tested image enhancing algorithms

Four real-time image enhancing algorithms were tested including: contrast limited adaptive histogram equalization (CLAHE), debluring (also known as unsharp mask), white balancing, and our newly designed method based on white balancing adapted to detection of markers that we call *marker-based underwater white balancing* – MBUWWB.

Contrast limited adaptive histogram equalization

Contrast limited adaptive histogram equalization (CLAHE) [PAA*87] is a variation of ordinary histogram equalization. Unlike ordinary histogram equalization, it works with a histogram of a small neighbor window around each pixel, and reduces the contrast of output image by clipping the highest values of input image histogram. CLAHE has a single parameter *clip limit*, which influences the amount of values clipped in the histogram. CLAHE was applied to Y channel, leaving U and V channels unchanged. For *clip limit*, we experimented with six values from 1 to 6.

Debluring

Deblur filter (or unsharp mask [KCFK17]) emphasizes high frequencies in input image by subtracting its low frequencies from itself. This filter is defined with the following equation:

$$I_{out} = (1+w) \cdot I_{in} - w \cdot \text{Gaussian}(I_{in}, \sigma) \quad (1)$$

where w represents the weight of subtracted low frequencies, and $\text{Gaussian}(I_{in}, \sigma)$ is a gaussian filter with standard deviation σ applied to the input image I_{in}. Deblur filter was also applied to Y channel, leaving U and V channels unchanged. We experimented with 16 combinations of values for σ and w: four values 1.0, 2.0, 3.0, and 4.0 for σ, and the same four values for w.

White balancing

White balancing (WB) transforms the colors in input image, so that white objects appeared as white under different illuminations (sun, clouded sky, or others). Although there are many sophisticated methods for white balancing, we used a method presented by Limare et al. [LLM*11]. This method was chosen because of its simplicity, processing speed, and universality.

foreach *color channel* **do**
  compute histogram of this channel;
  $channel_{min} \leftarrow black$-th percentile of values in histogram;
  $channel_{max} \leftarrow white$-th percentile of values in histogram;
  linearly transform all pixel intensities so that
    $channel_{min} = 0$ and $channel_{max} = 255$;
end

Algorithm 1: Pseudocode of white balancing algorithm

The algorithm is described by a pseudocode in Algorithm 1.

With values 0 for *black* and 100 for *white*, the algorithm transforms colors of the input image in a way that the colors of each color channel use the full range of possible values. This is similar to ordinary histogram equalization, but unlike histogram equalization, this transformation is done linearly without equalization. When choosing values higher than 0 for *black* and lower than 100 for *white*, the algorithm ignores the darkest and brightest pixels in the input image, which makes it more robust to noise. It should be noted that colors of restored images may not represent colors of objects properly, due to the simplicity of the algorithm. This is not a problem, however, since the restored image is not presented to the user; it is only processed by marker detectors. The white balancing algorithm is applied to all channels of RGB image, and then we convert the result into YUV space and use the Y channel for marker detection. We experimented with 16 combinations of values for percentiles *black* and *white*: four values 0, 1, 2, and 3 for percentile *black*, and four values 97, 98, 99, and 100 for percentile *white*.

Marker-based underwater white balancing

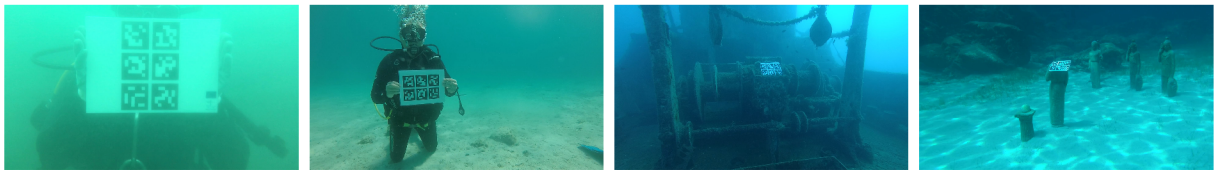
Marker-based underwater white balancing (MBUWWB) algorithm is our variation of the white balancing algorithm described in the previous section adapted to the problem of marker-based tracking in underwater environments. An intrinsic part of white balancing algorithms is to find colors that are subsequently mapped to the white and the black in the filtered image. White balancing algorithm described in the previous section chooses these colors as percentiles of values in input image histogram.

This behavior was adapted to a marker-based tracking. We assume that the marker is black and white, and instead of computing the histogram of the whole image, only the histogram of the part of the image which contains markers is computed. More precisely, if we detected any markers in the previous frame, we use this part of this previous frame with markers for computation of the histogram. If the previous frame did not contain any marker (or we process the first frame of the video), we compute the histogram of the whole image. This algorithm was applied to all channels of RGB image, and then the result is converted into YUV space, similarly as with ordinary white balancing algorithm. We also experimented with the same 16 combinations of values for percentiles *black* and *white*.

5. Visibility conditions

In this paper the main focus is on the detection of markers in different underwater visibility conditions. These conditions are influ-

© 2018 The Author(s)
Eurographics Proceedings © 2018 The Eurographics Association.



Environment 1	**Environment 2**	**Environment 3**	**Environment 4**
High level of turbidity	Moderate level of turbidity	Moderate level of turbidity	Low level of turbidity
Depth of 5 to 6 meters	Depth of 7 to 9 meters	Depth of 20 to 22 meters	Depth of 7 to 9 meters
iPad Pro 9.7-inch tablet	GoPro camera	GARMIN VIRB XE camera	NVIDIA SHIELD tablet
1920 × 1080, MPEG-2	1920 × 1080, MPEG-4	1920 × 1440, MPEG-4	1920 × 1080, MPEG-4
30 fps, 85 seconds	29.97 fps, 31 seconds	24 fps, 160 seconds	30 fps, 81 seconds

Figure 2: *Four different environments used in evaluation.*

enced mainly by turbidity of water, lighting conditions, presence of caustics, presence of small objects floating in the water, and also the size of the markers. All of these effects change the result of marker detection in differently with different impact. To cover all these effects, visibility in image is defined as follows. The computation of visibility is based on the number of detected markers. We compute *marker visibility* of each frame of our videos as a weighted average of the number of markers detected by all marker detectors and all image enhancing techniques. This average is used to obtain a value that is not biased towards any marker detector, any image processor, and any factor influencing conditions under water. Value of *marker visibility* can be expressed as:

$$marker\,visibility = \text{avg}_{det,proc}(w_{proc} * markers(det, proc(frame))) \tag{2}$$

where $\text{avg}_{det,proc}$ is an average took over all marker detectors *det* and image processors *proc*, w_{proc} is a weight of image processor *proc*, and $markers(det, proc(frame))$ is a number of markers detected by detector *det* in frame *frame* processed by image processor *proc*. Weight w_{proc} represents the weight of image processing algorithm. This weight was used, because we experimented with different number of parameter combinations for each image processing algorithm, and we computed the average of the number of detected markers of all parameter combinations. This weight equalizes all image processing algorithms, and is defined as

$$w_{proc} = 1/\#parameter\,combinations(proc) \tag{3}$$

where $\#parameter\,combinations(proc)$ is a number of combinations of parameters of image processor *proc*. This weight is therefore 1 for the results of detecting markers in the original unprocessed video, $1/6$ for CLAHE, and $1/16$ debluring, white balancing, and MBUWWB. Six markers are detected, so the value of *marker visibility* for given frame ranges from 0 (no marker is detected by any detector in the frame no matter what image processor is used) to 6 (all markers are detected by all detectors in the frame no matter what image processor is used). It is worth-mentioning that *marker visibility* was not used directly. 30 visibility levels were defined that represent 30 bins of values and have a range from 0 to 6 with an interval of 0.2. Each frame was assigned with one visibility level based on its *marker visibility*, and declare all frames with the same visibility level as frames with comparable visibility. The number of frames in each visibility level is shown in Figure 3.

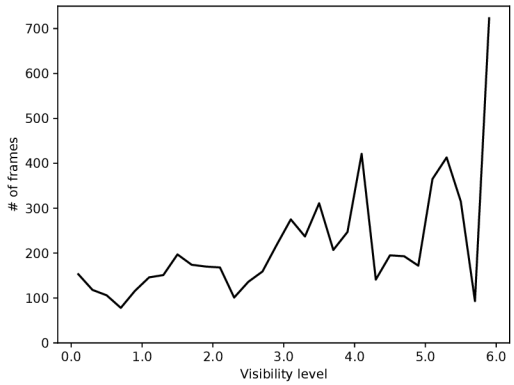


Figure 3: *Number of video frames with given visibility level. Frames without any markers (frames with zero marker visibility) are not considered in this graph.*

6. Results

Results are based on the number of correctly detected and identified markers in video frames enhanced by algorithms described in Section 4. Since no marker was recognized incorrectly if detected during tests, our comparisons do not examine false positives. All image enhancing algorithms and marker detection algorithms were compared to find the best parameters for each image enhancing algorithm, and then compared the results of all algorithms using these parameters. In addition to this, the processing and detection times of each algorithm was also measured.

Parameters of image enhancing algorithms

The average number of markers detected in frames of the same visibility level for each of 30 visibility levels was computed. This average was calculated separately for each marker detector and each combination of parameters of image enhancing algorithm. We present the results in supplementary materials. Instead of using the results for each visibility level, we compute an average of all these

© 2018 The Author(s)
Eurographics Proceedings © 2018 The Eurographics Association.

averages. This gives us overall number of detected markers that is not affected by a number of frames with given level of visibility. The results of this overall average for ARUco detector are in Table 1, the results for AprilTag2 detector are in Table 2.

Comparison of algorithms

Our results show that image enhancement improves marker detection performed by ARUco detector. Using the overall average results, ARUco detector found 2.460 markers in frames without any processing, and 2.679, 3.199, 3.133, and 3.429 markers in frames processed by CLAHE, Debluring, WB, and MBUWWB, respectively, when using the best combination of parameters. The results of AprilTag2 detector do not show any significant improvement. Without any image processing, AprilTag2 was able to find 3.278 markers. When the frames were processed by CLAHE, Debluring, WB, and MBUWWB, AprilTag2 was able to find 3.267, 3.295, 3.246, and 3.263 markers. The numbers of AprilTag2 are higher than most of the results of ARUco detector. This indicates that AprilTag2 is a superior marker detector when compared to ARUco (as was similarly observed by Cesar et al. [dSCGF*15]). Despite this, the highest number of detected markers was obtained by using ARUco detector and enhancing images with our MBUWWB algorithm. Based on these findings, we further compared the following combination of algorithms and parameters:

ARUco + Original ARUco detector without using any image enhancing method;

ARUco + CLAHE ARUco detector and CLAHE with clip limit 2;

ARUco + Debluring ARUco detector and debluring with $\sigma = 4.0$ and $w = 4.0$;

ARUco + WB ARUco detector and white balancing with $black = 2$ and $white = 99$;

ARUco + MBUWWB ARUco detector and MBUWWB with $black = 2$ and $white = 99$;

AprilTag2 + Debluring AprilTag2 detector and debluring with $\sigma = 4.0$ and $w = 1.0$ (we use only one combination of image enhancing algorithm and AprilTag2 detector, since other combinations reported similar results);

Figure 4 provides a comparison of selected algorithms using the average number of detected markers per visibility level. It is clearly shown that the last four combinations (*ARUco + Debluring*, *ARUco + WB*, *ARUco + MBUWWB*, and *AprilTag2 + Debluring*) provide the highest average number of detected markers, with *ARUco + MBUWWB* providing the highest numbers at most visibility levels. In this graph, we can also observe an unexpected drop in the number of detected markers of combinations *ARUco + Original* and *ARUco + CLAHE* between visibility levels 4.0 and 5.0. We assume that this behavior may be caused by lower number of evaluated frames in these visibility levels (see Figure 3) or by nature of evaluated marker detecting algorithm that stops finding markers well at this level of visibility.

Processing time

The time spent for detection of markers and for enhancing images was also computed. For the evaluation, a desktop PC with processor Intel Core i5 760, 8 GB of operating memory, and operating system

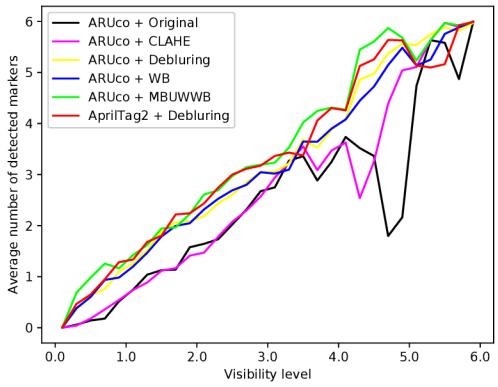


Figure 4: *Comparison of average number of detected markers per visibility level of the best combinations of marker detectors and image enhancing algorithms.*

Windows 10 was used. The measurement was performed using one video from **Environment 2**. To compare the performance of marker detecting algorithms, the time necessary to detect markers in the video without any image enhancement was measured. To compare the performance of image enhancing algorithms, the time required to enhance separate video frames, without performing any detection of markers was computed. As already mentioned, optimized implementations of ARUco and AprilTag2 algorithms were used. ARUco spent approximately 24.3 ms per frame, while AprilTag2 spent 246.9 ms per frame. It must be noted that the implementation of AprilTag2 is not optimized to use SIMD instructions, although we expect the computation time of an optimized version to be still much higher than the computation time of ARUco.

An implementation of CLAHE and debluring using functions available in OpenCV 3.2.0 was used. Since this library is optimized for using SIMD instructions and multiple threads, we implemented a similarly optimized version of white balancing and MBUWWB algorithms. The processing times were as follows: CLAHE 6.8 ms, debluring 28.2 ms, WB 7.0 ms, and MBUWWB 7.0 ms. The times were approximately the same for all parameters of these algorithms. In case of ARUco, the computation time of the whole combination with image pre-processing algorithms was also calculated. The parameters mentioned in the previous section was used again, and used the sum of the image preprocessing time and marker detection time. These are the following: CLAHE 41.8 ms, debluring 77.2 ms, WB 40.6 ms, and MBUWWB 32.1 ms.

7. Augmented reality reconstruction at Baiae

To assess the effectiveness of our results, a pilot underwater testing was performed in the sunken city called Baiae, which is located in Italy. Baiae contains buildings from ancient Rome, which submerged during the last 2000 years due to volcanic activity in this area. The focus of the testing was limited to one building, Villa a Protiro, with a characteristic mosaic in one of the rooms. Using the

© 2018 The Author(s)
Eurographics Proceedings © 2018 The Eurographics Association.

Video without enhancing	*avg*	**2.460**							
CLAHE	*clip limit*	1	2	3	4	5	6		
	avg	2.658	**2.679**	2.645	2.583	2.501	2.418		
Debluring	σ	1.0				2.0			
	w	1.0	2.0	3.0	4.0	1.0	2.0	3.0	4.0
	avg	2.528	2.539	2.532	2.504	2.630	2.747	2.821	2.877
	σ	3.0				4.0			
	w	1.0	2.0	3.0	4.0	1.0	2.0	3.0	4.0
	avg	2.681	2.847	2.946	3.053	2.730	2.933	3.082	**3.199**
WB	*black*	0				1			
	white	97	98	99	100	97	98	99	100
	avg	2.996	2.998	3.006	2.975	3.100	3.097	3.105	3.052
	black	2				3			
	white	97	98	99	100	97	98	99	100
	avg	3.128	3.120	**3.133**	3.086	3.106	3.109	3.113	3.080
MBUWWB	*black*	0				1			
	white	97	98	99	100	97	98	99	100
	avg	3.380	3.391	3.401	3.376	3.404	3.412	3.417	3.395
	black	2				3			
	white	97	98	99	100	97	98	99	100
	avg	3.405	3.412	**3.429**	3.388	3.409	3.420	3.420	3.391

Table 1: *Overall average number of markers* avg *detected by ARUco for each parameter of each image enhancing algorithm. This average is taken over averages per visibility level, so they are not affected by the number of frames with given level. The best result for each algorithm is emphasized.*

Video without enhancing	*avg*	**3.278**							
CLAHE	*clip limit*	1	2	3	4	5	6		
	avg	**3.267**	3.141	2.961	2.801	2.667	2.553		
Debluring	σ	1.0				2.0			
	w	1.0	2.0	3.0	4.0	1.0	2.0	3.0	4.0
	avg	3.141	2.980	2.806	2.640	3.200	3.067	2.928	2.768
	σ	3.0				4.0			
	w	1.0	2.0	3.0	4.0	1.0	2.0	3.0	4.0
	avg	3.264	3.205	3.131	3.064	**3.295**	3.276	3.225	3.170
WB	*black*	0				1			
	white	97	98	99	100	97	98	99	100
	avg	3.203	3.208	3.198	**3.246**	3.184	3.182	3.182	3.229
	black	2				3			
	white	97	98	99	100	97	98	99	100
	avg	3.145	3.148	3.148	3.200	3.070	3.079	3.079	3.143
MBUWWB	*black*	0				1			
	white	97	98	99	100	97	98	99	100
	avg	3.228	3.241	3.244	**3.263**	3.217	3.239	3.231	3.248
	black	2				3			
	white	97	98	99	100	97	98	99	100
	avg	3.209	3.217	3.216	3.223	3.194	3.190	3.187	3.198

Table 2: *Overall average number of markers* avg *detected by AprilTag2 for each parameter of each image enhancing algorithm. This average is taken over averages per visibility level, so they are not affected by the number of frames with given level. The best result for each algorithm is emphasized.*

© 2018 The Author(s)
Eurographics Proceedings © 2018 The Eurographics Association.

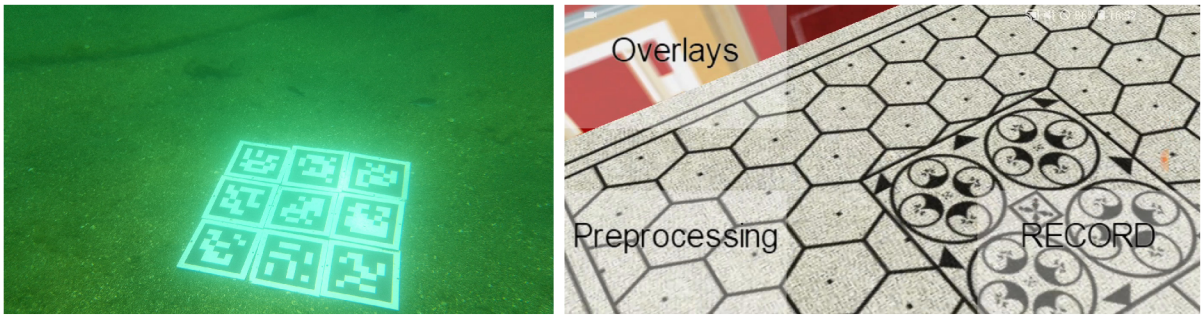


Figure 5: *Experiment performed at Baiae. In the left figure, we see markers placed at the location of the room with mosaic of Villa a Protiro. In the right figure, we see the reconstructed room in AR. With prefiltered image, we were able to detect more markers in more frames.*

Combination	# of frames with detected markers	# total number of detected markers
ARUco + Original	2888	15259
ARUco + CLAHE	3132	18865
ARUco + Debluring	3130	18864
ARUco + WB	3126	19041
ARUco + MBUWWB	3136	19407
AprilTag2 + Debluring	3141	19812

Table 3: *Results of the experiment conducted in Baiae, showing number of frames with at least one marker detected, and number of detected markers.*

improved approach, divers were able to perceive a 3D reconstruction of Villa a Protiro in AR. We used 9 markers from the same ARUco *DICT_6X6_50* dictionary, forming a grid of 3×3 markers. Size of each marker was 19 cm, and the space between markers was approximately 5 cm. A single video was recorded using Samsung S8 with resolution of 1920×1080 and length 141 seconds at 30 fps. Uncompressed images in NV21 format were stored as obtained from the camera. The setup and the application is illustrated in Figure 5.

The results of this experiment are shown in Table 3 and illustrate that AprilTag2 and ARUco with prefiltered input provides more images where at least one marker is detected, and therefore more images where the position of the camera can be calculated. Another interesting point is that ARUco detects the most markers when combined with MBUWWB, which provides us with more data for further processing, e.g., to calculate position of the camera more precisely. AprilTag2 provided slightly better results than ARUco combined with MBUWWB; however, the computation time is much larger, as mentioned in Section 6.

8. Discussion

Results indicate that proper choice of marker detection algorithm, which consists of steps like image preprocessing, thresholding, contour detection, and other, is very important for detecting markers in underwater environments. These environments heavily affect visibility, and thus they also affect the result of marker detection. It is not clear to us how exactly these conditions influence each detection step, but it was shown that the whole detection can be improved by adding another image preprocessing step. The combination of two fast algorithms, ARUco detector and our MBUWWB image enhancing algorithm, provided better results in less time when compared to more robust and much slower AprilTag2 in terms of the number of detected markers. This shows that proper image enhancing algorithm can replace complex preprocessing and thresholding methods in AprilTag2, and still obtain comparable results, as can also be seen in Figure 1.

Processing times showed us that different image enhancing algorithms influence the detection in different ways. Although debluring provided results comparable with our MBUWWB, images sharpened with debluring contain much more contours that must be rejected, which increases the detection time. Our combination of algorithms provided the result in a much smaller computation time than more robust solution. Also, when more markers creating a single multi marker are detected, we can compute a position of this multi marker with higher precision. Our experiment did not include any test to compare this precision, however, because in our experience, it is too difficult to obtain precise location of the marker in underwater environments to have a ground truth to which results could be compared.

Table 1 indicates that in case of the combination of ARUco and debluring algorithm, higher values of σ and w may lead to a higher number of detected markers. We performed additional tests with higher values of σ and w, and found that the highest overall average number of markers 3.550 was reached at $\sigma = 8.0$ and $w = 9.0$, which is comparable with our MBUWWB. However, the computation time also increased, due to larger gaussian kernel size that increases with σ and higher sharpness of features that increases with w. Processing time increased to 52.5 ms (image processing only) and 140.7 ms (image processing and marker detection), which makes it less usable for real-time applications when compared to MBUWWB.

Using more sophisticated algorithms for image enhancement is disputable, due to the time necessary to enhance images before detecting markers. Although some of these algorithms (especially those based on neural networks) provide results in very small time

© 2018 The Author(s)
Eurographics Proceedings © 2018 The Eurographics Association.

on PC, the processing power of mobile devices designed for augmented reality is lower, which reduces the number of usable algorithms. The results of white balancing methods indicate an important characteristic of computer vision performed in underwater environments. Luminance channel I, which is used in computer vision algorithms, is computed as a weighted sum of individual color channels R, G, and B:

$$I = w_r R + w_g G + w_b B$$

where w_r, w_g, and w_b are weights of these colors. However, since white balancing methods apply an affine transformation to each color channel individually, the white-balanced image can be obtain by using different weights to the original color channels. This indicates that a simple modification of the weights can improve the performance of computer vision algorithms in underwater environments, which can be done for free as a part of ordinal white balancing performed in digital cameras.

9. Conclusion

In this paper, we focused on the real-time detection of markers in underwater open sea environments that can be used for AR. We developed a method for enhancing images that is specialized to underwater environments and detection of markers. We compared this method with three methods for enhancing images on four sets of videos, each taken in a different environment. The results showed that our method combined with a fast marker detector gives better results than more sophisticated marker detector that runs much slower. In the future, a hybrid approach will be employed based on data generated by the visual tracking techniques illustrated in this paper and data from an acoustic modem. The solution will be integrated with a customized underwater tablet, which will estimate the position of the receiver by computing the distance from at least three fixed beacons placed on the seabed. The incoming positioning data from the various sensors will be finally processed through data fusion and error estimation algorithms.

Acknowledgments

This research was part of the i-MareCulture project (Advanced VR, iMmersive Serious Games and Augmented REality as Tools to Raise Awareness and Access to European Underwater CULTURal heritagE, Digital Heritage) that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 727153.

References

[AAHB12] ANCUTI C., ANCUTI C. O., HABER T., BEKAERT P.: Enhancing underwater images and videos by fusion. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Washington, DC, USA, 2012), IEEE Computer Society, pp. 81–88. 3

[AC14] ANCUTI C., CODRUTA A.: Effective contrast-based dehazing for robust image matching. 1871–1875. 2

[ADGS17] AGRAFIOTIS P., DRAKONAKIS G. I., GEORGOPOULOS A., SKARLATOS D.: The effect of underwater imagery radiometry on 3D reconstruction and orthoimagery. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLII-2/W3* (2017), 25–31. doi:10.5194/isprs-archives-XLII-2-W3-25-2017. 2

[BAC*16] BERGAMASCO F., ALBARELLI A., COSMO L., RODOLÁ E., TORSELLO A.: An accurate and robust artificial marker based on cyclic codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence 38* (2016), 2359–2373. doi:10.1109/TPAMI.2016.2519024. 3

[BBM09] BLUM L., BROLL W., MÜLLER S.: Augmented reality under water. In *SIGGRAPH '09: Posters* (New York, NY, USA, 2009), ACM. doi:10.1145/1599301.1599398. 2

[BKJ05] BENCINA R., KALTENBRUNNER M., JORDA S.: Improved topological fiducial tracking in the reacTIVision system. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops* (2005). doi:10.1109/CVPR.2005.475. 3

[BLB*16] BRUNO F., LAGUDI A., BARBIERI L., MUZZUPAPPA M., RITACCO G., COZZA A., COZZA M., PELUSO R., LUPIA M., CARIO G.: Virtual and augmented reality tools to improve the exploitation of underwater archaeological sites by diver and non-diver tourists. In *Digital Heritage. Progress in Cultural Heritage: Documentation, Preservation, and Protection: 6th International Conference, EuroMed 2016* (Cham, 2016), Springer International Publishing, pp. 269–280. doi:10.1007/978-3-319-48496-9_22. 2

[BLM*16] BRUNO F., LAGUDI A., MUZZUPAPPA M., LUPIA M., CARIO G., BARBIERI L., PASSARO S., SAGGIOMO R.: Project VISAS: Virtual and augmented exploitation of submerged archaeological site – overview and first results. In *Marine Technology Society Journal* (2016), vol. 50, pp. 119–129. 2

[BN15] BABAEE M., NEGAHDARIPOUR S.: Improved range estimation and underwater image enhancement under turbidity by opti-acoustic stereo imaging. In *OCEANS 2015* (2015), pp. 1–7. doi:10.1109/OCEANS-Genova.2015.7271611. 3

[BW13] BROWN H. C., WANG H.: Underwater augmented reality: Navigation and identification. In *2013 OCEANS - San Diego* (2013), pp. 1–5. 2

[CBME10] CARLEVARIS-BIANCO N., MOHAN A., EUSTICE R. M.: Initial results in underwater single image dehazing. In *OCEANS 2010* (2010), pp. 1–8. doi:10.1109/OCEANS.2010.5664428. 3

[CC12] CHIANG J. Y., CHEN Y.-C.: Underwater image enhancement by wavelength compensation and dehazing. *IEEE Transactions on Image Processing 21* (2012), 1756–1769. doi:10.1109/TIP.2011.2179666. 3

[CSK16] CHO Y., SHIN Y.-S., KIM A.: Online depth estimation and application to underwater image dehazing. In *OCEANS 2016 MTS/IEEE Monterey* (2016), pp. 1–7. doi:10.1109/OCEANS.2016.7761109. 3

[CXJ*16] CAI B., XU X., JIA K., QING C., TAO D.: Dehazenet: An end-to-end system for single image haze removal. *IEEE Transactions on Image Processing 25* (2016), 5187–5198. doi:10.1109/TIP.2016.2598681. 3

[DNBC16] DREWS P. L., NASCIMENTO E. R., BOTELHO S. S., CAMPOS M. F. M.: Underwater depth estimation and image restoration based on single images. *IEEE Computer Graphics and Applications 36* (2016), 24–35. doi:10.1109/MCG.2016.26. 3

[DNCE15] DREWS P., NASCIMENTO E. R., CAMPOS M. F. M., ELFES A.: Automatic restoration of underwater monocular sequences of images. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2015), pp. 1058–1064. doi:10.1109/IROS.2015.7353501. 3

[dSCGF*15] DOS SANTOS CESAR D. B., GAUDIG C., FRITSCHE M., DOS REIS M. A., KIRCHNER F.: An evaluation of artificial fiducial markers in underwater environments. In *OCEANS 2015* (2015), pp. 1–6. doi:10.1109/OCEANS-Genova.2015.7271491. 2, 3, 6

[Fat08] FATTAL R.: Single image dehazing. *ACM Trans. Graph. 27*, 3 (2008). doi:10.1145/1360612.1360671. 2

© 2018 The Author(s)
Eurographics Proceedings © 2018 The Eurographics Association.

[Fat14] FATTAL R.: Dehazing using color-lines. *ACM Trans. Graph. 34* (2014). doi:10.1145/2651362. 2

[Fia05] FIALA M.: Artag, a fiducial marker system using digital techniques. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* (Washington, DC, USA, 2005), IEEE Computer Society, pp. 590–596. doi:10.1109/CVPR.2005.74. 3

[Gal99] GALLAGHER D. G.: Development of miniature, head-mounted, virtual image displays for navy divers. In *OCEANS '99* (1999), vol. 3, pp. 1098–1104. doi:10.1109/OCEANS.1999.800143. 2

[GJnSMCMJ14] GARRIDO-JURADO S., NOZ SALINAS R. M., MADRID-CUEVAS F. J., MARÍN-JIMÉNEZ M. J.: Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition 47* (2014), 2280–2292. doi:http://dx.doi.org/10.1016/j.patcog.2014.01.005. 3

[GLW16] GAO Y., LI H., WEN S.: Restoration and enhancement of underwater images based on bright channel prior. 1–15. 2, 3

[HST11] HE K., SUN J., TANG X.: Single image haze removal using dark channel prior. *IEEE Trans. Pattern Anal. Mach. Intell. 33*, 12 (2011), 2341–2353. doi:10.1109/TPAMI.2010.168. 2, 3

[KB99] KATO H., BILLINGHURST M.: Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In *Augmented Reality, 1999. (IWAR '99) Proceedings. 2nd IEEE and ACM International Workshop on* (1999), pp. 85–94. doi:10.1109/IWAR.1999.803809. 3

[KCFK17] KRASULA L., CALLET P. L., FLIEGEL K., KLÍMA M.: Quality assessment of sharpened images: Challenges, methodology, and objective metrics. *IEEE Transactions on Image Processing 26* (2017), 1496–1508. doi:10.1109/TIP.2017.2651374. 2, 4

[KNC*08] KOPF J., NEUBERT B., CHEN B., COHEN M., COHEN-OR D., DEUSSEN O., UYTTENDAELE M., LISCHINSKI D.: Deep photo: Model-based photograph enhancement and viewing. *ACM Trans. Graph. 27*, 5 (2008). doi:10.1145/1409060.1409069. 2

[KPS10] KÖHLER J., PAGANI A., STRICKER D.: Robust detection and identification of partially occluded circular markers. In *VISAPP 2010 - Proceedings of the Fifth International Conference on Computer Vision Theory and Applications* (2010), pp. 387–392. 3

[LGG18] LI C., GUO J., GUO C.: Emerging from water: Underwater image color correction based on weakly supervised color transfer. *IEEE Signal Processing Letters 25* (March 2018), 323–327. doi:10.1109/LSP.2018.2792050. 3

[LLM*11] LIMARE N., LISANI J. L., MOREL J., PETRO A. B., SBERT C.: Simplest color balance. *IPOL Journal 1* (2011). doi:10.5201/ipol.2011.llmps-scb. 4

[LLZ*14] LI Y., LU H., ZHANG L., LI J., SERIKAWA S.: Real-time visualization system for deep-sea surveying. 1–10. 3

[LTT*15] LI Z., TAN P., TAN R. T., ZOU D., ZHOU S. Z., CHEONG L.-F.: Simultaneous video defogging and stereo reconstruction. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015), pp. 4988–4997. doi:10.1109/CVPR.2015.7299133. 2

[LW09] LEE J., WOO W.: Real-time color correction for marker-based augmented reality applications. In *International Workshop on Ubiquitous Virtual Reality 2009* (2009), pp. 32–35. 2

[MKMK09] MORALES R., KEITLER P., MAIER P., KLINKER G.: An underwater augmented reality system for commercial diving operations. In *Underwater Intervention Conference 2011* (2009), vol. 1, pp. 1–8. 2

[NF02] NAIMARK L., FOXLIN E.: Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self-tracker. In *Proceedings of International Symposium on Mixed and Augmented Reality* (2002), pp. 27–36. doi:10.1109/ISMAR.2002.1115065. 3

[OBLS13] OPPERMANN L., BLUM L., LEE J.-Y., SEO J.-H.: Areef multi-player underwater augmented reality experience. In *2013 IEEE International Games Innovation Conference (IGIC)* (2013), pp. 199–202. doi:10.1109/IGIC.2013.6659137. 2

[OBS16] OPPERMANN L., BLUM L., SHEKOW M.: Playing on areef: Evaluation of an underwater augmented reality game for kids. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services* (New York, NY, USA, 2016), ACM, pp. 330–340. doi:10.1145/2935334.2935368. 2

[OW04] OMER I., WERMAN M.: Color lines: Image specific color representation. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)* (2004), vol. 2, pp. 946–953. doi:10.1109/CVPR.2004.1315267. 2

[PAA*87] PIZER S. M., AMBURN E. P., AUSTIN J. D., CROMARTIE R., GESELOWITZ A., GREER T., ROMENY B. T. H., ZIMMERMAN J. B.: Adaptive histogram equalization and its variations. *Computer Vision Graph. Image Processing 39* (1987), 355–368. doi:10.1016/S0734-189X(87)80186-X. 4

[RLZ*16] REN W., LIU S., ZHANG H., PAN J., CAO X., YANG M.-H.: Single image dehazing via multi-scale convolutional neural networks. In *ECCV 2016* (2016), Springer International Publishing, pp. 154–169. doi:10.1007/978-3-319-46475-6_10. 3

[SAB*16] SKARLATOS D., AGRAFIOTIS P., BALOGH T., BRUNO F., CASTRO F., PETRIAGGI B. D., DEMESTICHA S., DOULAMIS A., DRAP P., GEORGOPOULOS A., KIKILLOS F., KYRIAKIDIS P., LIAROKAPIS F., POULLIS C., RIZVIC S.: Project imareculture: Advanced vr, immersive serious games and augmented reality as tools to raise awareness and access to european underwater cultural heritage. In *Digital Heritage* (2016), Springer International Publishing, pp. 805–813. 4

[SPOP16] SARAKINOU I., PAPADIMITRIOU K., OLGA G., PATIAS P.: Underwater 3D modeling: Image enhancement and point cloud filtering. 441–447. 2

[TATM13] TOYOURA M., ARUGA H., TURK M., MAO X.: Detecting markers in blurred and defocused images. In *2013 International Conference on Cyberworlds* (2013), pp. 183–190. doi:10.1109/CW.2013.58. 3

[TS09] TREIBITZ T., SCHECHNER Y. Y.: Polarization: Beneficial for visibility enhancement? In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2009), pp. 525–532. doi:10.1109/CVPRW.2009.5206551. 2

[Tsi16] TSIRIKOLIAS K. D.: Low level image processing and analysis using radius filters. *Digital Signal Processing 50* (2016), 72–83. doi:https://doi.org/10.1016/j.dsp.2015.12.001. 2

[Wei06] WEISS B.: Fast median and bilateral filtering. *ACM Trans. Graph. 25*, 3 (2006), 519–526. doi:10.1145/1141911.1141918. 2

[WO16] WANG J., OLSON E.: AprilTag 2: Efficient and robust fiducial detection. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (October 2016). 3

[ŽČB*18] ŽUŽI M., ČEJKA J., BRUNO F., SKARLATOS D., LIAROKAPIS F.: Impact of dehazing on underwater marker detection for augmented reality. *Frontiers in Robotics and AI 5* (2018), 1–13. doi:10.3389/frobt.2018.00092. 2

[ZH14] ZHANG J., HU S.: A gpu-accelerated real-time single image dehazing method using pixel-level optimal de-hazing criterion. *Journal of Real-Time Image Processing 9* (2014), 661–672. doi:10.1007/s11554-012-0244-y. 2

[ZMS15] ZHU Q., MAI J., SHAO L.: A fast single image haze removal algorithm using color attenuation prior. *IEEE Transactions on Image Processing 24* (2015), 3522–3533. doi:10.1109/TIP.2015.2446191. 2

© 2018 The Author(s)
Eurographics Proceedings © 2018 The Eurographics Association.

Article

Detecting Square Markers in Underwater Environments

Jan Čejka [1,*], Fabio Bruno [2], Dimitrios Skarlatos [3] and Fotis Liarokapis [4]

[1] Human Computer Interaction Laboratory, Faculty of Informatics, Masaryk University, Botanická 68a, Brno, Czech Republic; xcejka2@fi.muni.cz
[2] 3D Research s.r.l. — University of Calabria, Rende, 87036 Cosenza, Italy; fabio.bruno@unical.it
[3] Photogrammetric Vision Laboratory, Department of Civil Engineering and Geomatics, Cyprus University of Technology, 3036 Limassol, Cyprus; dimitrios.skarlatos@cut.ac.cy
[4] Human Computer Interaction Laboratory, Faculty of Informatics, Masaryk University, Botanická 68a, Brno, Czech Republic; liarokap@mail.muni.cz
* Correspondence: xcejka2@fi.muni.cz

Version January 16, 2019 submitted to Remote Sens.

Abstract: Augmented reality can be deployed in various application domains, such as enhancing human vision, navigation, manufacturing, medicine, military, entertainment and archaeology. One of the least explored areas is underwater environment. The main benefit of augmented reality in these environments is that it can help divers with orientation or present interesting information for archaeological and touristic purposes (i.e. non-existing buildings or shipwrecks before sinking). However, harsh sea environment affects algorithms of computer vision, and complicates detection of object, which is essential for augmented reality. This paper presents a new algorithm for detection of fiducial markers that is tailored for underwater environments. It also proposes a method that generates synthetic images with such markers in these environments. This new detector is compared with existing solutions using synthetic images and images taken in real world, showing that it performs better that other detectors – it finds more markers than faster algorithms, and runs faster than robust algorithms that detect the same amount of markers.

Keywords: Augmented reality; marker-based tracking; generating synthetic images; real-time; cultural heritage;

1. Introduction

Historical objects and places play important roles in lives of people, since history is of great significance for every culture. Old buildings are repaired, protected, and preserved for future generations. Archaeologists search for artifacts to recover data about our history. They connect pieces of information together to obtain solid basis of our culture. Sites of archaeological significance are not only on land, but can be located under water. Typical examples are ships that sank into water with the cargo they transported, or cities that submerged due to earthquakes or tsunami waves. Their remains are recovered to understand about the culture of the sailors or learn about the history of the sites. During the past decade, people started diving to experience an underwater view of these sites. There are cases where old ships are even intentionally sunk into water to become a tourist attraction for divers [1].

Nowadays, technology has evolved and provides new options and opportunities to present the past to people. Advances in the field of mobile devices allow us to use them for augmented reality (AR) applications to see in real time what cannot be seen in the real world, and through a screen of smart phones or tablets, it is possible to visualise in three-dimensions how buildings looked hundreds or thousand years ago [2,3].

One of the most crucial things that are necessary for AR applications is to know the precise pose (position and orientation) of the device and all important objects around. In urban environments, there are lots of available solutions such as sensors (GPS, Wi-Fi, etc.), or computer vision solutions. Unfortunately, the situation is different under water. In the sea, there is no GPS signal or other signals to get the location of the device, and also due to worsen visibility conditions, objects that are away further than a dozen of meters cannot be distinguished. If water is very turbid, the visibility drops to a few meters. For these reasons, it is still very hard to recognize objects under water and track their position, so it is necessary to populate the underwater environment with either sensors (e.g., acoustic beacons) or recognizable objects (e.g., markers) to give divers a robust solution to see the history of underwater places in AR.

This paper is focused on AR applications in underwater environments and especially on detection of square markers that are used to obtain the pose of mobile devices and objects under water. The main contributions of this paper are:

- a new method for synthetically generated images of markers in underwater conditions;
- a new algorithm for detection of square markers that is adapted for underwater environments;
- comparison of this algorithm with other state-of-the-art algorithms on synthetically generated images and on real images.

After presenting a state of the art in the field of detecting markers, Section 2 presents the new algorithm and algorithms with which it is compared. Then, Section 3 and Section 4 present the method for generating synthetic images, experiments done with these images, and results of these experiments. Similarly, Section 5 and Section 6 describe the videos taken under water and results of tests performed with these videos. The final conclusion of this paper is in Section 7.

1.1. Related work

Square markers are one of the most common method for pose estimation in AR applications. They are easy and fast to detect, and contain enough information to compute the relative pose to the camera. One of the first software libraries that offered this functionality is ARToolKit [4], or its newer version ARToolKitPlus optimized to run especially on mobile devices [5]. To distinguish between square markers, this library uses an arbitrary image that is placed inside the marker. Square markers are also used in ARTag library [6], which is also focused on a description of individual markers. This library uses a 6×6 binary pattern located inside each marker, and is able to correct several bits of wrongly detected code using redundancy information. ARUco [7] and its improved version ARUco3 [8] are libraries for detection of square markers based on OpenCV. They also uses a 6×6 binary pattern like ARTag, but unlike ARTag, they design a dictionary of patters specifically to be able to correct the largest number of error bits. Another detectors, AprilTag [9] or its newer version AprilTag2 [10], provide robust implementation of square marker detectors that are resistant to lighting conditions and steep angles of view.

Other types of markers are also used in AR. Circular or elliptic markers [11][12] can be used to compute the position of these markers even when the markers are partially occluded, since elliptic shape of their contours provides more information about its position than in case of square markers. A price for this is a higher processing time. RUNE tag [13] defines markers that are made of unconnected dots, and therefore it does not detect any contours like the previous solutions. Other markers are also possible: markers of irregular shapes are described in [14] and used in reacTIVision library; markers described in [15] are designed specifically to be detectable in blurred and unfocused images; and Fourier Tags markers [16] that are also designed to be used in non-ideal conditions and encode their information in frequency domain.

Markers simplify the problem of detecting 3D objects, since they are 2D and designed to be easily recognized. However, it is also possible to perform this detection and recognition without markers, using distinctive image features like edges, corners, or textures. Such algorithms consist of two parts:

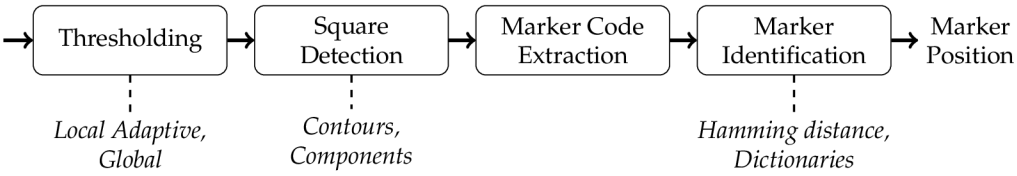


Figure 1. General workflow of algorithms detecting square markers

detection of these features, and computation of a descriptor for each feature that is able to match the features between frames. One of the first algorithm for natural feature detection used in augmented reality was SIFT [17][18]. This algorithm uses difference of Gaussians to detect interesting features in the image, and describes each feature with a feature vector of 128 floating point computed from gradient magnitudes around it. SURF detector [19] uses approximation of Hessian to detect important features, and builds its descriptor on a response of Haar wavelets. Both of these solutions use floating point arithmetic, which is more demanding than integer arithmetic, but there are also solutions that use mainly integer operations and binary comparisons to detect and register features. FAST [20] is a corner detector that uses machine learning approach to find the fastest order of comparisons, and since it performs only comparisons, it runs very fast. This algorithm itself does not describe the features, but can be used in combination with feature descriptors. BRIEF [21] is a descriptor that assigns each feature a vector of 128 bits representing a binary result of 128 comparisons of pixels around a feature. Since it is a binary vector, matching features is faster than in case of SIFT and SURF. BRISK [22] is a natural feature detector that uses AGAST [23] (variation of FAST) to detect features, and compares pixels in circular patters around features to obtain rotation and scale invariance. FREAK [24] is a detector similar to BRISK, but its circular pattern is based on human retina. ORB detector [25] is composed of FAST detector and BRIEF descriptor that are optimized to provide robust detection invariant to rotations and scales. Natural feature detectors that run fast can also be used in simultaneous localization and mapping (SLAM) solutions, e.g., ORB-SLAM [26] or LSD-SLAM [27].

The problem of improving images taken in bad visibility conditions for the purpose of improving computer vision results was examined in several scenarios. Agarwal et al. [28] compares methods for increasing contrast of images taken in smoke occluded environments to improve visual odometry based on a sequence of RGB-D images. Cesar et al. [29] evaluates the performance of marker detecting algorithms in artificially simulated bad underwater conditions. Performance on marker detecting and image improving algorithms was also evaluated by Žuži et al. [30] on videos taken in sea. Several authors focused on registration of images, and based their comparisons on the number of detected and matched SIFT features. Andono et al. [31] compared contrast enhancing techniques for matching a sequence of underwater images. Ancuti et al. presents an algorithm removing haze in foggy images [32] and another algorithm for correcting colors in underwater images [33], both for the purpose of matching images taken in different views. Gao et al. [34] develops a method for restoring colors of underwater images, and compares it with other methods using a number of detected SIFT features and a number of detected Canny edges. Image improving methods were also used to improve the results of reconstructed underwater objects using photogrammetry in the work of Agrafiotis et al. [35] or Marino et al. [36].

2. Marker detection and image improving algorithms

This section presents algorithms that are used further in experiments. First, it describes marker detecting algorithms ARUco, ARUco3, and AprilTag2, then image improving algorithms CLAHE, Debluring, White Balancing, and Marker-Based Underwater White Balancing, and finally our new marker detecting algorithm UWARUco.

2.1. ARUco, ARUco3, and AprilTag2

Marker detecting algorithms ARUco, ARUco3, and AprilTag2 follow general structure described in Figure 1. First, incoming image is thresholded using global or local adaptive methods to obtain simplified binary image. This image is in the second step searched for square polygons that become candidates for markers. In the third step, identification code is extracted from inner area of each candidate, and the candidate is then in the fourth step identified and discarded if it does not belong to a set of searched codes. Finally, the position of the marker is computed from its corners or edges.

ARUco detector [7] was designed to run fast and reliably recognize marker codes. It uses adaptive algorithm to threshold input image that compute the threshold value for each pixel as an average of surrounding pixels. To detect marker-like shapes, it finds all contours, and filter out those that do not represent square polygons (small contours, non-polygonal contours, etc.). After the squares are detected, it unprojects them to remove perspective distortion, thresholds them again (this time using Otsu threshold), and obtains the inner code, which is checked with a dictionary to remove errors. If it is correct, the corners of these squares are used to compute their relative position to the camera.

ARUco3 detector [8] is based on ARUco detector and was improved to run fast with high-resolution images. There are two main differences between this algorithm and ARUco. The first difference is that this algorithm exchanges adaptive thresholding with simpler global thresholding, which is faster, but less robust to uneven lighting. The second difference is that it scales the image down, so that its size is still sufficient to detect and recognize markers, but the detection runs faster than in the original image. Implementation provided by authors exists in three versions: Normal, Fast, and VideoFast. Normal version uses adaptive thresholding like ARUco, and therefore should be more robust to uneven illuminance. Fast version uses global thresholding, and applies Otsu method on an image part with a marker to compute the threshold for the next frame (or if no marker is available, it chooses the threshold randomly). VideoFast version is like Fast version, but additionally it assumes that markers in a frame have approximately the same size as the markers in the previous frame and optimizes the scaling accordingly.

AprilTag2 detector [10] also uses an adaptive algorithm for thresholding, but instead of using an average of all surrounding pixels, it searches these pixels for the lowest and highest intensities, and chooses the threshold as the average between these two intensities. Then, it segments the binary result, fit quads, recover codes, and uses a hash table to check if the code is correct and the quad is a marker.

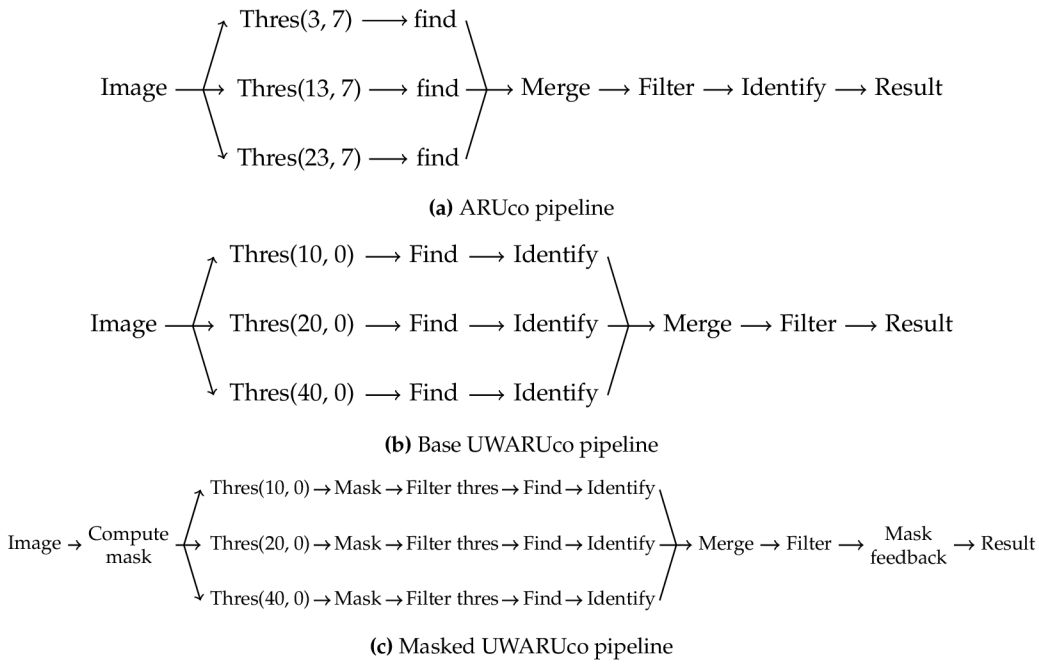
2.2. Real-time algorithms improving underwater images

Authors in [37] discuss a possibility of using real-time image improving algorithms to increase the number of detected markers. They compare four algorithms: CLAHE, deblur, white balancing (WB), and marker-based underwater white balancing (MBUWWB); these algorithms are also compared in this paper.

Contrast Limited Adaptive Histogram Equalization (CLAHE) [38] is based on equalizing image histograms. At each pixel, it computes a histogram of its surroundings, rearanges it to avoid unnatural changes in contrast, and equalizes it to obtain new intensity. Deblur [39] (also known as debluring or unsharp mask filter) stresses edges by removing low frequencies from the original image, which can be described by the following equation:

$$Image^{out} = (1 + w) \cdot Image^{in} - w \cdot Gaussian(Image^{in}, \sigma_{deblur}) \tag{1}$$

White balancing algorithms changes the colors in image to look more natural. There is a lot of white balancing algorithms, but for performance reasons, authors in [37] chose a simple algorithm from [40]. This algorithm computes a histogram of input image, removes $perc_{dark}$ percent of darkest pixels and $100 - perc_{bright}$ percent of brightest pixels, and changes the colors to linearly stretch the rest of it. Its adaptation for marker-based tracking, marker-based underwater white balancing, described also in [37], does the same, but computes the initial histogram only of areas that contain markers.



(a) ARUco pipeline

(b) Base UWARUco pipeline

(c) Masked UWARUco pipeline

Figure 2. Detailed description of ARUco and its underwater adaptations, Base version of UWARUco, and Masked version of UWARUco.

Many of these algorithms have parameters that influence their behaviour. In our experiments, we used parameters from [37]: 2 as a *clip limit* of CLAHE, 4 as both σ_{deblur} and w for debluring, and 2 as $perc_{dark}$ and 99 as $perc_{bright}$ for white balancing and marker-based underwater white balancing.

2.3. Detection of markers under water

Underwater ARUco (UWARUco) is an adaptation of ARUco algorithm for underwater environments. Pipeline of the original ARUco algorithm is depicted Figure 2a. It takes input grey-scale image, obtains a thresholded image in *Thres* step, and find contours in this image in *Find* step. These threshold and find steps are done three times in parallel threads, each time with different sizes of a window that computes the mean value used as the threshold (default sizes are 3, 13, and 23 pixels). Before actual thresholding, the threshold is increased by a constant value to suppress contours created by small noise (default value of this constant is 7). After the contours are found in each thresholded image, the result is merged and filtered in *Merge* and *Filter* steps to represent each marker with only one contour no matter in how many thresholded images the contour is found, and finally in *Identify* step, the original image is thresholded again, now using Otsu method, to obtain marker code, which is then identified.

We analyzed results of [37] to investigate, which steps are influenced by image improving algorithms when the number of detected markers is increased. It was found that all four tested image improving algorithms increase the contrast of the image, and additionally, deblur also acts as another blur applied to the image before thresholding. Similar effect can be achieved by increasing threshold window size, and decreasing the constant that is added before thresholding, see Figure 2b. Experiments with different values showed that optimal results are obtained with the constant set to 0 and window sizes 10, 20, and 40, see step *Thres*. However, changing parameters of the thresholding method is not enough, since image improving methods also affect the image that is thresholded by Otsu method when marker code is being extracted. This issue is solved by using the image that was thresholded in *Thres* step instead of performing an additional threshold, and moving *Identify* step between *Find* step and *Merge* step. We call this method a *Base* version of *UWARUco*.

Input : Grey scale image $Image^{in}$ whose mask is to be computed, threshold for brightness mask *brightness_threshold*, threshold for noise mask *noise_threshold*
Output: Brightness and noise masks *brightness_mask* and *noise_mask*
$mins \leftarrow$ image of one fourth of the size of $Image^{in}$ with minimums of 4×4 pixels of $Image^{in}$;
$maxs \leftarrow$ image of one fourth of the size of $Image^{in}$ with maximums of 4×4 pixels of $Image^{in}$;
$mins \leftarrow erode(mins)$;
$maxs \leftarrow dilate(maxs)$;
$diffs \leftarrow maxs - mins$;
$brightness_mask \leftarrow maxs \geq brightness_threshold$;
$noise_mask \leftarrow diffs \geq noise_threshold$;
Algorithm 1: Pseudocode of algorithm for computing brightness mask and noise mask.

Input : *maxs* and *diffs* from computing masks, upscaled to the size of original image, contours of all detected markers *contours*
Output: Thresholds *brightness_threshold* and *noise_threshold*.
if *contours is empty* **then**
  $brightness_threshold \leftarrow 0$;
  $noise_threshold \leftarrow 0$;
else
  brightness_threshold $\leftarrow$ 80% of minimum of *maxs* at pixels of *contours*;
  noise_threshold $\leftarrow$ 80% of minimum of *diffs* at pixels of *contours*;
end
Algorithm 2: Pseudocode of algorithm for computing thresholds for brightness mask and noise mask.

This algorithm detected more markers in underwater videos and original ARUco, as will be shown in Section 4 and Section 6. However, its processing time was slower, due to an increase in number of contours that are found in thresholded images and must be processed by the detector. To improve the processing speed of the detection, this algorithm is extended with a binary mask and a filter that is applied on the thresholded image before the contours are detected, to simplify the binary image, reduce the number of contours, and decrease the computation time. We call this algorithm a *Masked* version of *UWARUco*, see Figure 2c.

The mask was developed for detecting parts of image that do not contain markers, and is composed of two submasks: brightness masks, and noise mask. Brightness mask detects parts of image with very high brightness, because the brightest pixels are often white areas of markers, since underwater environments do not contain a lot of white objects. Noise mask removes areas that do not contain strong edges, and thus keeps only those parts that represents edges of markers and removes parts with noise. Computation of both submasks is performed in *Compute mask* step, and is described using pseudocode in Algorithm 1. First, the minimum and maximum intensities are computed for blocks of 4×4 pixels, which results in images *mins* and *maxs* that are in each dimension four times smaller than the original image. These images are blurred by spilling minima and maxima of each pixel into its neighbourhood by taking minimum or maximum of surrounding pixels, which is achieved by computing morphological erosion and dilatation. Without this blur, the resultant thresholded image do not contain areas where 4×4 blocks are separated by marker contour, and thus one of these blocks is bright and the other is dark. After this blur, *maxs* image is thresholded and used for brightness mask, and difference image *diffs*, computed by subtracting *mins* from *maxs*, is thresholded to obtain noise mask.

Computation of thresholds for these masks is based on areas with detected markers, and is done after the detection in *Mask feedback* step. The procedure is described in Algorithm 2. Images *maxs* and *diffs* that are used to compute masks in Algorithm 1 are inspected at pixels of contours of found markers, and the lowest values of these pixels are used as thresholds. Such values would be sufficient as thresholds for the current frame, but to handle small changes in lighting conditions between frames,

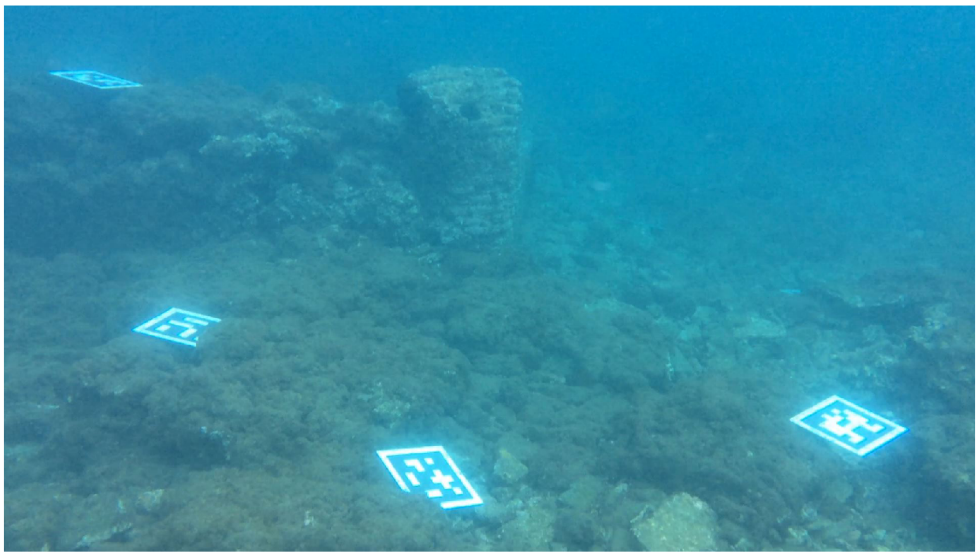


Figure 3. Effect of turbidity and glow in a real image. Due to a turbidity, distant objects are worse visible than close objects, similarly as in foggy images. Also, notice a small glow around the central and the right marker.

they are decreased by 20 % to obtain actual thresholds for the following frame (this value was chosen after several experiments with different constants).

These two submasks are applied on the thresholded image in UWARUco step *Mask*. In addition to this, the thresholded and masked binary image is filtered in *Filter thres* step to remove small contours that appear due to the presence of a noise. The filtering is done by applying 3×3 median filter on the thresholded image, because contours are composed of just a few pixels. Since the input image is binary, the median can also be obtained by computing a mean of surrounding pixels and thresholding it to a threshold of 0.5.

3. Generating synthetic images

Our synthetic generator of images considers the following effects that appear under water: turbidity of water, glowing of markers, sensor noise, and a blur caused by not having objects perfectly focused. The effect of turbidity and glow in real world is depicted in Figure 3.

Effect of turbidity simulates a decrease in visibility of distant objects, which is caused by very small particles dispersed in water. This effect is very close to an effect of fog, and for this reason, in the rest of the paper we refer to this effect as to the effect of fog. Small particles in water and post-processing effects in cameras also make markers glowing, which happens when white areas of markers reflect a high amount of light (for example, when the marker is facing up), and this light is blurred due to the fog. This glow often makes white areas of the marker overexposed and black marker areas having higher intensity than marker background. Sensor noise consists of thermal noise and other types of noise that appear when shooting images. We consider only thermal noise, and simulate it with Gaussian random noise, as other types of noises can be either simulated as a part of this Gaussian noise, or are not very significant. Effect of blur caused by objects that are not perfectly focused is added to avoid unnaturally sharp color changes of markers in images, and is represented by Gaussian blur with a small sigma.

Pipeline of this image-generating algorithm is presented in Figure 4. It consists of five main steps that change the generated image (*Background*, *Drawing Markers*, *Out-of-focus Blur*, *Glow Blur*, and *Sensor Noise*), and three supporting steps that compute the parameters for main steps (*Glow Effect*, *Fog Effect*, and *Perspective Effect*).

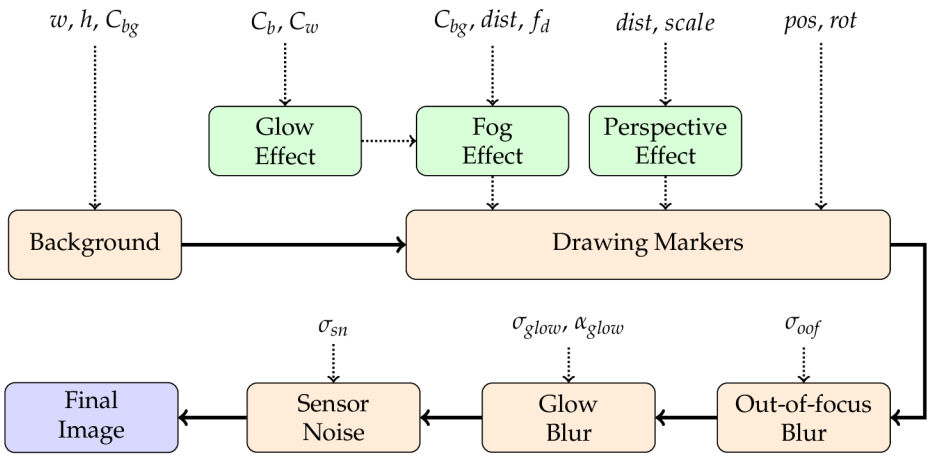


Figure 4. Pipeline of synthetic image generator. See the text for further details.

The first step is *Background* step. In this step, the image is allocated with width w and height h (in our experiments, we used images in Full HD resolution – 1920×1080 pixels), and filled with background color C_{bg}. In *Drawing Markers* step, six markers are generated and rendered onto the background image with intensities C_b and C_w representing black and white marker regions. Each marker contains a 6×6 pixels of binary code, one pixel of black border, and additional one pixel of white border, as if the marker was printed on a white square paper. Then, they are rotated with an angle *rot*, scaled by a factor *scale*, placed at position *pos* to form a grid of two rows and three columns, and further displaced by up to 40 pixels.

Parameters *scale*, C_b, and C_w are changed by supporting steps *Glow Effect*, *Fog Effect*, and *Perspective Effect*. In *Perspective Effect* step, basic *scale* is divided by the distance of the marker from the camera *dist*, so the markers that are further away are smaller. Intensities C_b and C_w are first increased in *Glow effect* step, and then mixed with background intensity C_{bg} in *Fog effect* step. The increase in intensity done by the glow effect depends on the experiment, but generally, this increase can be arbitrary. The mix done by the fog effect is based on standard exponential fog, which depends on the distance from camera *dist*, and uses the following equations:

$$C_b^{out} = C_b^{in} \cdot e^{-f_d \cdot dist} + C_{bg} \cdot (1 - e^{-f_d \cdot dist}) \tag{2}$$

$$C_w^{out} = C_w^{in} \cdot e^{-f_d \cdot dist} + C_{bg} \cdot (1 - e^{-f_d \cdot dist}) \tag{3}$$

The third step called *Out-of-focus Blur* blurs the image very little to remove sharp changes in colors, which is done with a simple Gaussian blur with sigma σ_{oof} (in all our images, this sigma was set to 1.0). Glow effect not only changes the intensities of black and white areas of the marker, but also blurs the image. This blurred image is obtained by applying a Gaussian blur with sigma σ_{glow}, which is then mixed with the original image with factor α_{glow} to preserve sharp edges, as described in the following equation:

$$Image^{out} = (1 - \alpha_{glow}) \cdot Image^{in} + \alpha_{glow} \cdot Gaussian(Image^{in}, \sigma_{glow}) \tag{4}$$

In the last step, *Sensor Noise* step, the image is changed by adding a sensor noise represented by Gaussian noise with zero mean and sigma σ_{sn} (note that the intensities of pixels can be decreased as well as increased in this step). The intensities of pixels are then clamped to range 0 and 255, and written to the final image. Example of an image generated by this algorithm is presented in Figure 5.

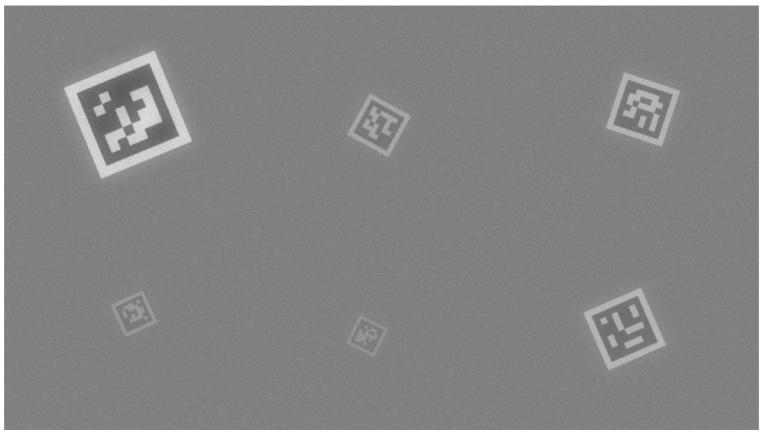


Figure 5. Synthetically generated image

4. Experiments with synthetically generated images

Synthetically generated images were used in five experiments: a reference experiment with good conditions, an experiment simulating bad visibility conditions, an experiment simulating markers at different depths in foggy conditions, an experiment simulating glowing markers, and an experiment with all effects applied together. Experiments were performed on a desktop PC with processor Intel Core i5 760, 8 GB of operating memory, and operating system Windows 10, and image generating algorithm was implemented using functions of OpenCV 3.4.3.

Processing time was computed only from frames that contained identified markers and followed after another frame with identified markers. This procedure provides better results regarding performance of algorithms that utilize data from previous frames, because these algorithms are often optimized for such situations. If there are no such frames, the result is missing, as visible in graphs.

4.1. Reference with good visual conditions

The first experiment evaluated a basic performance of marker detectors in very good visual conditions to obtain a reference for other results. In this test, fog effect, glow effect, perspective effect, and sensor noise effect were not applied, the intensity of the background was set to 150, intensities of black and white marker colors were set to 0 and 255 respectively, and the scale of the markers was a variable parameter in this experiment, changing from 5 to 30 pixels. Intensity of the background was chosen to be a bit lighter that half-tone intensity to resemble real images, but does not influence the results, as observed in preliminary tests.

Each value of scale was tested in 20 videos consisting of 100 frames, the results (percentage of detected markers and detection time) were averaged and are presented in Figure 6. It is shown that good visibility conditions are not a problem for any of the solution, although some solutions have problems with specific scales of markers. ARUco detector starts to lose some markers when the scale of the marker is higher than 23 – the same value as is the maximum window size of its adaptive thresholding algorithm. When the scale is larger, thresholded image contains a fictional contour inside marker border, and in some cases, this contour is chosen as the contour of the marker, which makes the detected marker smaller and often not identified. This behaviour is emphasized by some image improving algorithms.

Normal version of ARUco3 struggles to find small markers. Further tests found that this is caused by white border around markers, which is too small, but when its size is increased, the number of detected markers reaches 100 %. The processing times show that Fast and VideoFast variants of ARUco3 are clearly the fastest algorithms for detection of markers in these very good visual conditions.

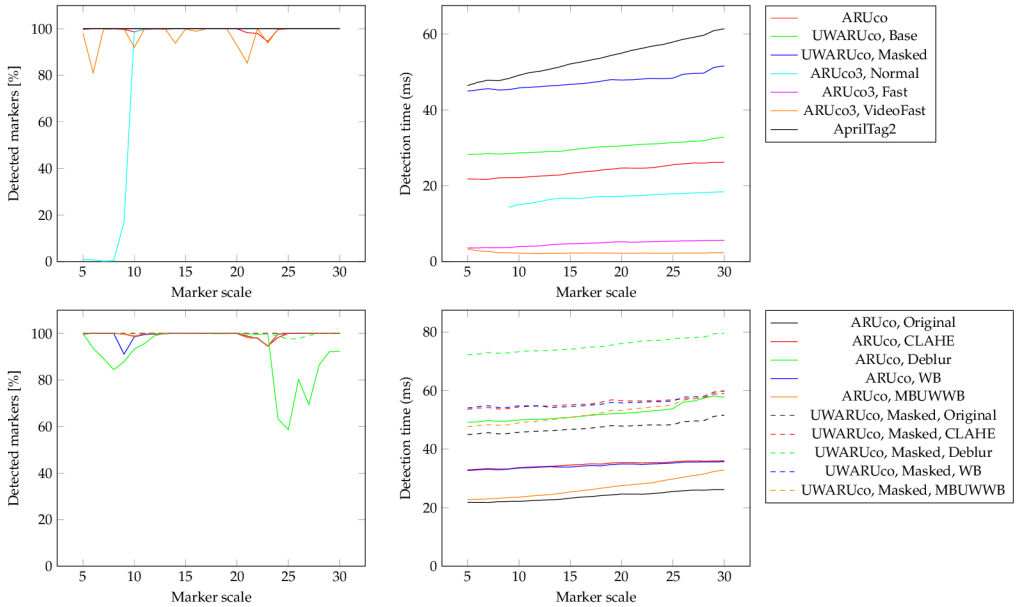


Figure 6. Results of reference test. Top row: results of marker detecting algorithms; bottom row: results of image improving algorithms; left column: percentage of detected markers; right column: detection time in milliseconds.

Normal version of ARUco3 and ARUco provide the results in reasonable speeds, while the speed of UWARUco and ARUco combined with image improving algorithms is lower, since these algorithms contain additional processing steps. The speed of AprilTag2 is also reasonably low in this test.

None of the solutions had problems with precision of detection, although this precision decreased with larger markers in case of ARUco and UWARUco and their variants, because these detectors chose wrong contours in some cases. However, even in these cases, corners of markers were found with precision of units of pixels.

4.2. Bad visibility conditions

The task here was to simulate bad visibility conditions that cause lower contrast of the image. Glow and perspective effects were not applied, intensity of background was again set to 150, and the scale of all markers was set to 15 to prevent ARUco and UWARUco from choosing wrong contours, as observed in the previous experiment. Sigma of sensor noise σ_{sn} was set to 5, which creates images that are similar to real images. Effect of fog was applied, but it was modified for this test. Since this effect changes only the difference between intensities of black and white regions of the marker, this difference became the main parameter of this analysis. Instead of depending on the distance *dist* or fog density f_d, intensities of black and white are set to specific values, starting at 70 and 230 to simulate low-foggy environment, and moving towards the intensity of the background, ending at 148 and 152 to simulate high-foggy environment. The difference between these intensities therefore starts with 160 and ends with 4.

The results are presented in Figure 7. In worse visibility conditions, Fast and VideoFast versions of ARUco3 require more trials to find proper global threshold values, but when they do, they do not loose them, unless the visibility conditions are very bad. The decrease in the number of detected markers that is visible in the graphs is caused by lost of markers in the first video frames. This experiment also shows that UWARUco is able to find markers in very bad conditions, being the last algorithm that detects markers when the difference between black and white regions gets very low. AprilTag2 is also

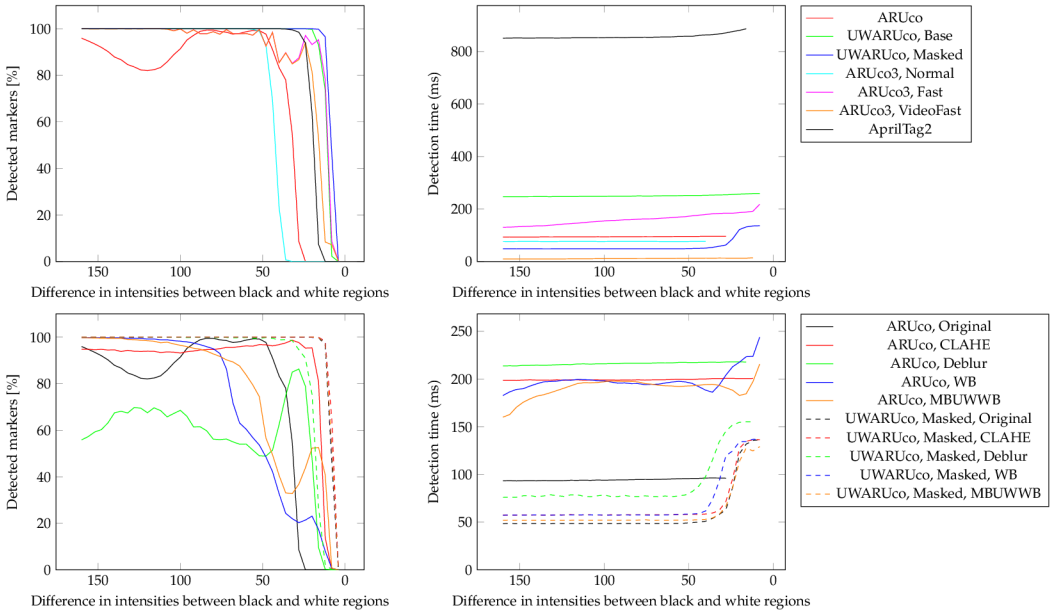


Figure 7. Results of test with bad visibility. Top row: results of marker detecting algorithms; bottom row: results of image improving algorithms; left column: percentage of detected markers; right column: detection time in milliseconds. Since the detection time is evaluated only when the markers are detected, there are no values when markers are lost.

able to find markers bad conditions, and so are Fast and VideoFast versions of ARUco3 if they choose a good global threshold, but their results are worse than results of UWARUco.

Image improving algorithms did not performed well in this test. The results show that there is no need to improve images for UWARUco, and in case of ARUco, the improvement in the number of detected markers is disputable, since the number of detected markers often increased in bad visibility conditions, but decreased in good visibility conditions.

The processing time is higher when compared to the times obtained in the reference, mainly due to a presence of sensor noise in the image. The processing time of AprilTag2 increases very much, which makes this algorithm unusable in real-time solutions. The processing times of ARUco, Base version of UWARUco, and Normal and Fast versions of ARUco3 also increases, but less than then time of AprilTag2. The processing time of the Masked version of UWARUco remained the same as in the reference, so it become faster than other algorithms, whose processing time increased. The fastest solution of this test is the VideoFast version of ARUco3, but it detects less markers.

4.3. Foggy conditions and markers at different distances

Here, markers were placed in foggy environment at different distances from the viewer, which influenced their size and also their contrast, due to the presence of fog. This distance was set between values 1.0 and 4.0, and changed continuously between successive frames, since abrupt changes could negatively influence marker detectors that use data from previous frames. The scale of the markers started at 30 at distance 1.0 and decreased linearly with the distance, intensity of the background was again fixed at 150, the intensities of black and white marker regions were 0 and 255 at zero distance, the glow effect was not present, and the sigma of sensor noise σ_{sn} was again set to 5. Density of the fog was set to $f_d = 1.0$, so that the difference in intensities between black and white regions at maximum depth was approximately 5, similarly as in the previous test when choosing the worst conditions.

This test has no variable parameter, so only one set of 20 videos with 100 frames was generated. The results are presented in Table 1. Masked version of UWARUco detected the highest number

Solution	ARUco	UWARUco		ARUco3			AprilTag2
		Base	Masked	Normal	Fast	VideoFast	
Detected markers [%]	32.067	64.658	75.050	24.000	60.392	45.450	51.342
Detection time [ms]	96.002	255.633	117.907	76.769	89.064	9.564	1220.796

Solution	ARUco				
	Original	CLAHE	Deblur	WB	MBUWWB
Detected markers [%]	32.067	63.225	43.808	46.025	57.525
Detection time [ms]	96.002	201.421	218.623	200.801	193.417

Solution	UWARUco, Masked				
	Original	CLAHE	Deblur	WB	MBUWWB
Detected markers [%]	75.050	79.525	58.500	77.992	77.358
Detection time [ms]	117.907	128.644	141.740	128.436	117.376

Table 1. Results of test with markers at different distances

of markers, while its processing time is comparable with other algorithms, with an exception of VideoFast version of ARUco3, which is faster, but detects lower number of markers. The results of image improving algorithms show that in this test, they were able to increase the number of detected markers of ARUco at a cost of higher detection time, some of them placing ARUco at approximately the same level as Base version of UWARUco. They also slightly improved the results of Masked version of UWARUco, again at a cost of increased processing time.

4.4. Glowing markers

Next, the detection of markers in foggy environments that appear to glow in the image was evaluated. Perspective and fog effects were not applied, scale of markers was set to 15, and sigma of sensor noise was set to $\sigma_{sn} = 5$. The intensity of the background was set to a low value 50, so that black marker areas could have higher intensity that the background. The glow effect was simulated with 20 levels, each level increasing the intensity of the black marker areas linearly from 0 to 100, intensity of white marker areas from 255 to 400, glow alpha α_{glow} from 0.0 to 0.8, and glow sigma σ_{glow} from 0 to 60.

Similarly to the previous experiments, each glow effect level was tested with 20 videos consisting of 100 frames. The averaged results are presented in Figure 8. Several marker detecting algorithms (ARUco, Fast and VideoFast versions of ARUco3, and AprilTag2) struggle at finding markers when the glow was at high levels – Fast and VideoFast variants of ARUco3 again lose several frames until they found proper global threshold value. The processing time of all detectors stayed approximately the same without any influence caused by increased level of glow. Since the number of detected markers was already very high, image improving algorithms could not increase these values. Some of them (CLAHE and deblur combined with ARUco) decreased the number of detected markers, but many of them just increased the computation time. One exception to this is the combination of ARUco and MBUWWB, which decreased the computation time at glow levels above 6 while keeping the number of detected markers very high. This happened because at glow levels higher than 6, black marker regions have higher intensity than background, so most of the artificial sensor noise is removed, which makes the thresholded image mostly noise-free.

4.5. All effects

Finally, all effects affecting synthetically generated images were incorporated. Several variables changed at the same time, continuously between successive frames: intensity of background C_{bg} was changing between values 50 and 200, distance of markers *dist* between values 1.0 and 4.0, and intensity of fog f_d between values 0 and 1. Intensity of glow effect was also changing, simultaneously influencing intensity of black and white regions of markers, glow alpha, and glow sigma. Intensity of black regions C_b was changing between values 0 and 100, intensity of white regions C_w between values 255 and 400, glow alpha α_{glow} between values 0.0 and 0.8, and glow sigma σ_{glow} between values

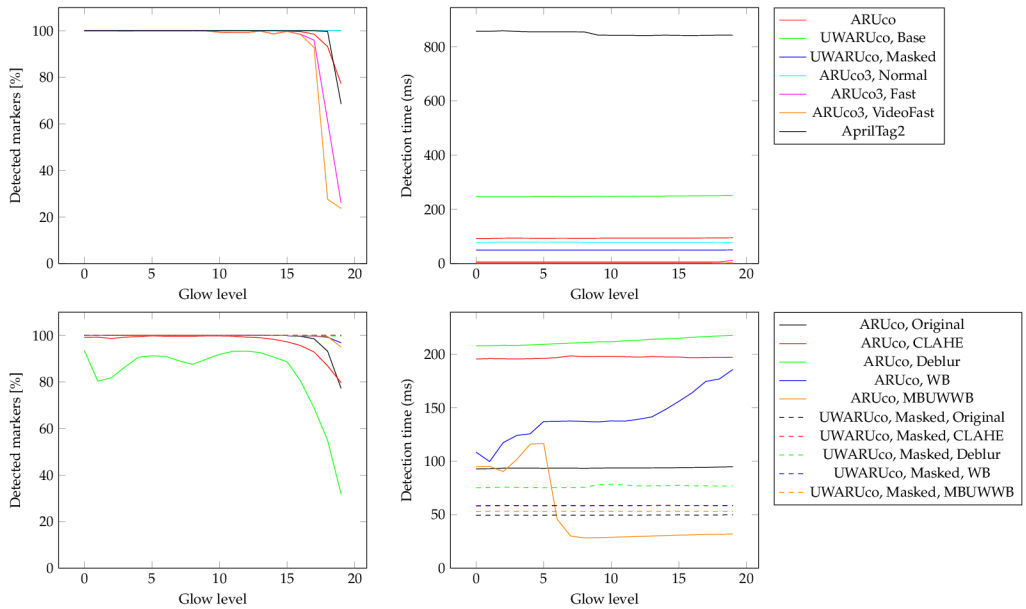


Figure 8. Results of test with glowing markers. Top row: results of marker detecting algorithms; bottom row: results of image improving algorithms; left column: percentage of detected markers; right column: detection time in milliseconds.

0 and 40. Similarly as in the previous experiments, the scale of markers was kept same at value 30 at distance 1, and sigma of sensor noise σ_{sn} at value 5.

This test also has no variable parameter, so only one set of 20 videos with 100 frames was generated. The results are presented in Table 2. The highest number of detected markers was obtained by both versions of UWARUco, with Masked version detecting at times lower than other algorithms (except for ARUco3). This shows that UWARUco, especially its Masked version, is best suited for underwater environments. All versions of ARUco3 algorithm detected the lowest number of markers, approximately 50 %, though their detection time was very low, especially in case of VideoFast version. Results of image improving techniques are rather surprising, because in previous experiments, these methods gave promising results only when detecting markers at different distances, but in other tests, they decreased the number of detected markers. Here, these methods increased the number of detected markers, although again at increased time of detection. This result is similar to that obtained in [37] when tested on real images.

Solution	ARUco	UWARUco		ARUco3			AprilTag2
		Base	Masked	Normal	Fast	VideoFast	
Detected markers [%]	64.542	92.808	95.200	53.392	53.025	49.925	80.417
Detection time [ms]	96.558	257.800	88.623	77.213	32.207	5.888	846.431

Solution	ARUco				
	Original	CLAHE	Deblur	WB	MBUWWB
Detected markers [%]	64.542	81.467	53.575	74.733	74.925
Detection time [ms]	96.558	201.264	219.608	194.947	153.949

Solution	UWARUco, Masked				
	Original	CLAHE	Deblur	WB	MBUWWB
Detected markers [%]	95.200	95.492	88.450	94.850	94.317
Detection time [ms]	88.623	102.573	121.527	88.171	79.625

Table 2. Results of test with all effects active

4.5.1. Discussion on synthetic images

Synthetic generation of images is able to test many situations that appear in underwater environments, however, some of these situations were not tackled. One limitation of generated images is that all markers are facing the camera, but this problem of skewed markers is not bound solely to underwater environments, and requires a construction of a specialized marker detector to deal with it. In our experience with real images, the detection of markers is also influenced by objects that obscure markers, be it floating objects like fish, or static objects like surrounding rocks and plants. Similarly to the previous issue, solution to this problem requires a specialized marker detector.

All tested images also have a single-color background, which does not appear in real images, since the background is made of various objects. The algorithm for generating images can be improved by swapping the background with an image or a video that is taken under water, but this solution requires a lot of videos in different environments with well assessed parameters for effect, to fit the generated markers seamlessly into images and to prevent the results from being shifted towards specific videos.

Results of the tests do not mention a number of falsely detected markers. Unless we include the results with mischosen contours as described in Section 4.1, falsely detected markers appeared extremely rarely. From 176000 frames processed by 15 combinations of marker detecting and image improving algorithms, only 111 falsely detected marker appeared, detected by ARUco (1), ARUco with CLAHE (10), ARUco with deblur (11), ARUco with WB (11), ARUco with MBUWWB (7), Masked version of UWARUco (5), Masked version of UWARUco with CLAHE (3), Masked version of UWARUco with deblur (4), Masked version of UWARUco with WB (3), Masked version of UWARUco with MBUWWB (6), and AprilTag2 (50). Base version of UWARUco and all versions of ARUco3 did not report any false positive.

5. Materials and methods of underwater experiments

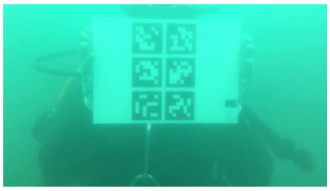
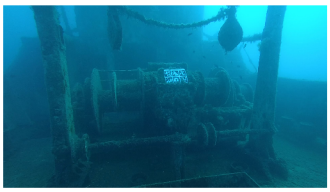
Marker detecting solutions were also compared on images taken under water in Mediterranean sea. This experiment reported only the number of detected markers and the processing time, and not the precision of the detection, for the following reasons. First, the divers were unable to measure the location of markers with a sufficient precision, which requires to be very high, since the experiments with synthetic images found that the ground truth location of markers must be known with a precision of units of pixels. Second, due to a high number of frames, bad visibility conditions, and required precision, the ground truth was infeasible to obtain manually.

Parameters of videos are shown in Table 3. This table describes a level of turbidity in the videos, depth under sea at which the videos were taken, device that took the video, resolution of the video, compression algorithm, frame rate, and length in seconds. Sets **Baiae1**, **Athens**, **Constandis**, and **Green Bay** were taken from [41], set **Villa** was taken from [37], and sets **Baiae2**, and **Epidauros** are new. It should be noted that the level of visibility is not based on any measurement, but it was chosen by authors by assessing the videos.

Sets **Baiae1**, **Athens**, **Constandis**, and **Green Bay** contain a board with six markers from ARUco library, each with size 8 cm. Set **Athens** also contained a single separate marker with size 16 cm, and sets **Villa** and **Epidauros** contained 9 separate markers of size 19 cm, placed into a grid of 3 times 3 markers. In set **Baiae2**, separate markers of size 19 cm were placed isolated from each other on a ground around other objects. Measuring was performed on the same computer that run experiments with synthetically generated videos. Computation time was also evaluated only on frames with detected markers if they followed another frame with detected markers.

6. Results of underwater tests

Total number of detected markers and average processing time are shown in Table 4. ARUco is able to detect markers in videos with medium visibility (**Villa**, **Baiae2**), but struggles to detect markers in poor visibility conditions (**Baiae1**). The results are better when using image improving techniques,

	
Nm: **Baiae1** Lc: Baiae, Italy Tr: High Dp: 5 – 6 meters Dv: iPad Pro 9.7inch Rs: 1920×1080 Cm: MPEG-2 FL: 30 fps, 85 s	Nm: **Athens** Lc: Athens, Greece Tr: Moderate Dp: 7 – 9 meters Dv: GoPro camera Rs: 1920×1080 Cm: MPEG-4 FL: 30 fps, 31 s
	
Nm: **Constandis** Lc: Limassol, Cyprus Tr: Moderate Dp: 20 – 22 meters Dv: GARMIN VIRB XE Rs: 1920×1440 Cm: MPEG-4 FL: 24 fps, 160 s	Nm: **Green Bay** Lc: Green Bay, Cyprus Tr: Low Dp: 7 – 9 meters Dv: NVIDIA SHIELD Rs: 1920×1080 Cm: MPEG-4 FL: 30 fps, 81 s

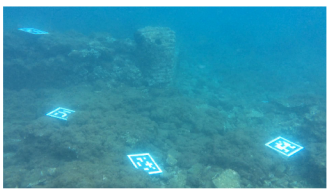
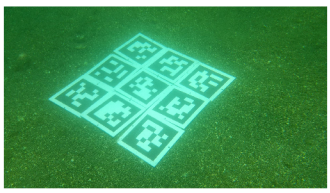
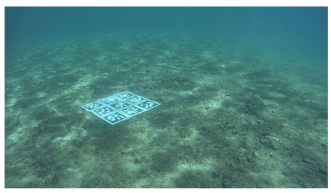
		
Nm: **Villa** Lc: Villa a Protiro, B., It. Tr: Moderate Dp: 5 – 6 meters Dv: Samsung Galaxy S8 Rs: 1920×1080 Cm: no compression FL: 30 fps, 141 s	Nm: **Baiae2** Lc: Baiae, Italy Tr: Moderate Dp: 5 – 6 meters Dv: iPad Mini 2 Rs: 1920×1080 Cm: MPEG-4 FL: 30 fps, 421 s	Nm: **Epidauros** Lc: Epidauros, Greece Tr: Low Dp: 4 – 6 meters Dv: Sony FDR-X1000V Rs: 3840×2160 Cm: MPEG-4 FL: 24 fps, 180 s

Table 3. Evaluated sets of videos. Abbreviations: Nm: name of the set; Lc: location where the set was taken; Tr: assessed level of turbidity; Dp: depth at which the set was taken; Dv: recording device; Rs: resolution of the video; Cm: compression method; FL: frame rate and length in seconds. Location *B., It.* stands for *Baiae, Italy*.

however, no technique provides the best result in all sets. MBUWWB provides good results in **Villa** and especially in **Baiae1**, but it struggles with detection of markers in set **Baiae2**, where the markers are at different distances from the viewer, which also means their visibility is different. This represents a complication for MBUWWB, because this technique is a global techniques (as well as other techniques,

Solution		ARUco	UWARUco		ARUco3			AprilTag2
			Base	Masked	Normal	Fast	VideoFast	
Baiae1	# of markers	467	6004	6223	36	2145	1893	5695
	Time [ms]	24.627	111.528	57.680	15.397	6.970	1.748	178.874
Athens	# of markers	5272	5913	5832	4877	5565	2610	5923
	Time [ms]	31.202	86.503	53.771	21.398	6.550	2.843	240.672
Constandis	# of markers	6981	6944	6878	6578	5382	4511	6332
	Time [ms]	39.010	139.514	65.565	25.030	6.672	5.446	327.887
Green Bay	# of markers	7964	7932	7594	7062	6347	5784	7332
	Time [ms]	27.531	74.712	49.915	18.040	10.537	5.288	222.003
Villa	# of markers	14457	19879	20145	9947	14316	12589	20082
	Time [ms]	75.747	140.659	62.872	58.266	6.700	2.368	323.005
Baiae2	# of markers	13829	15932	14976	12466	8257	3869	14577
	Time [ms]	34.552	83.614	55.093	21.535	4.563	1.672	230.910
Epidauros	# of markers	18749	25126	25713	13864	10480	6340	21628
	Time [ms]	181.441	425.286	252.327	152.590	23.575	4.692	1252.122

Solution		ARUco				
		Original	CLAHE	Deblur	WB	MBUWWB
Baiae1	# of markers	467	3646	3857	4098	5140
	Time [ms]	24.627	38.064	80.560	43.370	47.475
Athens	# of markers	5272	5504	5611	5640	5616
	Time [ms]	31.202	48.424	86.524	45.867	38.271
Constandis	# of markers	6981	6279	6850	6737	6869
	Time [ms]	39.010	73.406	136.165	86.436	53.236
Green Bay	# of markers	7964	6406	8355	7392	7896
	Time [ms]	27.531	49.000	84.394	50.442	35.355
Villa	# of markers	14457	18981	18958	19029	19398
	Time [ms]	75.747	138.263	178.143	114.494	45.239
Baiae2	# of markers	13829	14126	11445	11500	10227
	Time [ms]	34.552	62.603	100.112	50.598	27.755
Epidauros	# of markers	18749	20030	23597	16929	21696
	Time [ms]	181.441	341.750	586.785	296.562	181.483

Solution		UWARUco, Masked				
		Original	CLAHE	Deblur	WB	MBUWWB
Baiae1	# of markers	6223	6612	6081	6129	5605
	Time [ms]	57.680	64.902	93.618	64.998	55.866
Athens	# of markers	5832	5806	5848	5831	5829
	Time [ms]	53.771	60.742	85.729	60.054	52.622
Constandis	# of markers	6878	6702	6959	6749	6843
	Time [ms]	65.565	75.938	105.121	76.404	68.263
Green Bay	# of markers	7594	6776	7464	7149	7599
	Time [ms]	49.915	56.289	78.405	56.515	51.390
Villa	# of markers	20145	20281	19400	20225	20258
	Time [ms]	62.872	82.574	114.332	65.231	61.647
Baiae2	# of markers	14976	15996	15356	12988	13244
	Time [ms]	55.093	65.498	85.929	62.256	55.415
Epidauros	# of markers	25713	26012	23618	21991	25623
	Time [ms]	252.327	305.263	388.771	274.608	247.548

Table 4. Results of detecting markers in real underwater videos

e.g., WB), and it changes the whole image in the same manner. ARUco3 detector also does not produce consistent results, the table shows that in case of **Villa**, Fast version of this detector detects the highest number of markers and Normal version finds the lowest number of markers, but an opposite result is found in **Baiae2** and other sets. It must be noted, however, that both Fast and VideoFast versions are also global techniques, and exhibit similar problems as MBUWWB image improving technique. AprilTag2 is able to detect a very high number of markers in most situations, however, its processing

time is also very high, and therefore hard to use in real-time augmented reality applications, especially when targeting mobile devices.

Base version of UWARUco detects consistently a very high number of markers. This is compensated with its processing time, which is also higher – it is higher than the processing times of original ARUco and all versions of ARUco3, approximately the same as the processing times of slower combinations of ARUco and image improving algorithms, and still lower than processing times of AprilTag2. However, the results of Masked version of UWARUco show that the number of detected markers can still be very high while the processing time is much lower. They also show that using additional image improving techniques is not necessary, although using some of them can still lead to better results at a price of increased processing time.

6.1. Discussion on underwater experiments

Comparison between results of Base and Masked versions of UWARUco shows that the number of detected markers can increase when the thresholded image is masked before detecting contours. This unexpected outcome happens because this mask does not remove contours, instead it merges them into larger contours, which can connect borders of markers that are disconnected in original image, and therefore detecting a marker that is not detected in the original image.

According to [8], ARUco3 is designed to run fast with high-resolution images at a price of reducing its adaptability to worse visibility and lighting conditions. This is confirmed in the results – Fast and VideoFast versions of ARUco3 provide the fastest processing times when compared to other methods, but the number of detected markers drops in worse visibility conditions. The number of false positives was not counted in this experiment, since the ground truth solution is missing. However, it is expected to be very low, similarly as in the previous test with synthetically generated images.

7. Conclusions

This paper presented a new algorithm for detecting square markers in underwater environments, called UWARUco, which is based on ARUco marker detector, but replaces its method for reducing a number of contours caused by noise with a method that masks out areas that do not contain any markers. It performs better in underwater environments, it detects more markers than preceding version of the algorithm, and provides the result faster than state-of-the-art robust solutions. This allows AR applications to run more fluently and recognize objects from further distances in worse visibility conditions.

Future work will be focused on fusion of this algorithm with other methods that are used to compute and maintain orientation of a mobile device. Using techniques like Kalman filter, the position obtained visually by detecting markers will be fused with results of internal device sensors like accelerometer or gyroscope, and devices that are able to provide absolute position underwater like acoustic beacons. This will create a sophisticated solution for underwater augmented reality, which can be deployed in applications that present underwater cultural heritage to general public, or help divers with orientation in deep underwater archaeological sites.

Author Contributions: Conceptualization, Jan Čejka; Data curation, Fabio Bruno and Dimitrios Skarlatos; Investigation, Jan Čejka; Methodology, Jan Čejka and Fotis Liarokapis; Software, Jan Čejka; Writing – original draft, Jan Čejka; Writing – review & editing, Fabio Bruno, Dimitrios Skarlatos and Fotis Liarokapis.

Funding: This research is a part of the i-MareCulture project (Advanced VR, iMmersive Serious Games and Augmented REality as Tools to Raise Awareness and Access to European Underwater CULTURal heritagE, Digital Heritage) that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 727153.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Edney, J.; Spennemann, D.H.R. Can Artificial Reef Wrecks Reduce Diver Impacts on Shipwrecks? The Management Dimension. *Journal of Maritime Archaeology* **2015**, *10*, 141–157. doi:10.1007/s11457-015-9140-5.
2. Vlahakis, V.; Ioannidis, N.; Karigiannis, J.; Tsotros, M.; Gounaris, M.; Stricker, D.; Gleue, T.; Daehne, P.; Almeida, L. Archeoguide: An Augmented Reality Guide for Archaeological Sites. *IEEE Computer Graphics and Applications* **2002**, *22*, 52–60. doi:10.1109/MCG.2002.1028726.
3. Panou, C.; Ragia, L.; Dimelli, D.; Mania, K. An Architecture for Mobile Outdoors Augmented Reality for Cultural Heritage. *ISPRS International Journal of Geo-Information* **2018**, *7*. doi:10.3390/ijgi7120463.
4. Kato, H.; Billinghurst, M. Marker Tracking and HMD Calibration for a Video-Based Augmented Reality Conferencing System. Augmented Reality, 1999. (IWAR '99) Proceedings. 2nd IEEE and ACM International Workshop on, 1999, pp. 85–94. doi:10.1109/IWAR.1999.803809.
5. Wagner, D.; Schmalstieg, D. ARToolKitPlus for Pose Tracking on Mobile Devices. 2007.
6. Fiala, M. ARTag, a Fiducial Marker System Using Digital Techniques. Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05); IEEE Computer Society: Washington, DC, USA, 2005; pp. 590–596. doi:10.1109/CVPR.2005.74.
7. Garrido-Jurado, S.; noz Salinas, R.M.; Madrid-Cuevas, F.J.; Marín-Jiménez, M.J. Automatic Generation and Detection of Highly Reliable Fiducial Markers Under Occlusion. *Pattern Recognition* **2014**, *47*, 2280–2292. doi:http://dx.doi.org/10.1016/j.patcog.2014.01.005.
8. Romero-Ramirez, F.J.; noz Salinas, R.M.; Medina-Carnicer, R. Speeded up detection of squared fiducial markers. *Image and Vision Computing* **2018**, *76*, 38–47. doi:https://doi.org/10.1016/j.imavis.2018.05.004.
9. Olson, E. AprilTag: A robust and flexible visual fiducial system. 2011 IEEE International Conference on Robotics and Automation, 2011, pp. 3400–3407. doi:10.1109/ICRA.2011.5979561.
10. Wang, J.; Olson, E. AprilTag 2: Efficient and robust fiducial detection. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2016.
11. Naimark, L.; Foxlin, E. Circular Data Matrix Fiducial System and Robust Image Processing for a Wearable Vision-Inertial Self-Tracker. Proceedings of International Symposium on Mixed and Augmented Reality, 2002, pp. 27–36. doi:10.1109/ISMAR.2002.1115065.
12. Köhler, J.; Pagani, A.; Stricker, D. Robust Detection and Identification of Partially Occluded Circular Markers. VISAPP 2010 - Proceedings of the Fifth International Conference on Computer Vision Theory and Applications, 2010, pp. 387–392.
13. Bergamasco, F.; Albarelli, A.; Cosmo, L.; Rodolá, E.; Torsello, A. An Accurate and Robust Artificial Marker Based on Cyclic Codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2016**, *38*, 2359–2373. doi:10.1109/TPAMI.2016.2519024.
14. Bencina, R.; Kaltenbrunner, M.; Jorda, S. Improved Topological Fiducial Tracking in the reacTIVision System. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops, 2005. doi:10.1109/CVPR.2005.475.
15. Toyoura, M.; Aruga, H.; Turk, M.; Mao, X. Detecting Markers in Blurred and Defocused Images. 2013 International Conference on Cyberworlds, 2013, pp. 183–190. doi:10.1109/CW.2013.58.
16. Xu, A.; Dudek, G. Fourier Tag: A Smoothly Degradable Fiducial Marker System with Configurable Payload Capacity. 2011 Canadian Conference on Computer and Robot Vision, 2011, pp. 40–47. doi:10.1109/CRV.2011.13.
17. Lowe, D.G. Object recognition from local scale-invariant features. Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999, Vol. 2, pp. 1150–1157. doi:10.1109/ICCV.1999.790410.
18. Gordon, I.; Lowe, D.G. Scene modelling, recognition and tracking with invariant image features. Third IEEE and ACM International Symposium on Mixed and Augmented Reality, 2004, pp. 110–119. doi:10.1109/ISMAR.2004.53.
19. Bay, H.; Tuytelaars, T.; Van Gool, L. SURF: Speeded Up Robust Features. Computer Vision – ECCV 2006; Leonardis, A.; Bischof, H.; Pinz, A., Eds.; Springer Berlin Heidelberg: Berlin, Heidelberg, 2006; pp. 404–417.
20. Rosten, E.; Drummond, T. Machine Learning for High-Speed Corner Detection. Computer Vision – ECCV 2006; Leonardis, A.; Bischof, H.; Pinz, A., Eds.; Springer Berlin Heidelberg: Berlin, Heidelberg, 2006; pp. 430–443.

21. Calonder, M.; Lepetit, V.; Strecha, C.; Fua, P. BRIEF: Binary Robust Independent Elementary Features. Computer Vision – ECCV 2010; Daniilidis, K.; Maragos, P.; Paragios, N., Eds.; Springer Berlin Heidelberg: Berlin, Heidelberg, 2010; pp. 778–792.
22. Leutenegger, S.; Chli, M.; Siegwart, Y. Brisk: Binary robust invariant scalable keypoints. In Computer Vision (ICCV), 2011 IEEE International Conference on, 2011, pp. 2548–2555.
23. Mair, E.; Hager, G.D.; Burschka, D.; Suppa, M.; Hirzinger, G. Adaptive and Generic Corner Detection Based on the Accelerated Segment Test. Computer Vision – ECCV 2010; Daniilidis, K.; Maragos, P.; Paragios, N., Eds.; Springer Berlin Heidelberg: Berlin, Heidelberg, 2010; pp. 183–196.
24. Ortiz, R. FREAK: Fast Retina Keypoint. Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); IEEE Computer Society: Washington, DC, USA, 2012; pp. 510–517.
25. Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G. ORB: An efficient alternative to SIFT or SURF. 2011 International Conference on Computer Vision, 2011, pp. 2564–2571. doi:10.1109/ICCV.2011.6126544.
26. Mur-Artal, R.; Montiel, J.M.M.; Tardós, J.D. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics* **2015**, *31*, 1147–1163. doi:10.1109/TRO.2015.2463671.
27. Engel, J.; Schöps, T.; Cremers, D. LSD-SLAM: Large-Scale Direct Monocular SLAM. Computer Vision – ECCV 2014; Fleet, D.; Pajdla, T.; Schiele, B.; Tuytelaars, T., Eds.; Springer International Publishing: Cham, 2014; pp. 834–849.
28. Agarwal, A.; Maturana, D.; Scherer, S. Visual Odometry in Smoke Occluded Environments. Technical report, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 2014.
29. dos Santos Cesar, D.B.; Gaudig, C.; Fritsche, M.; dos Reis, M.A.; Kirchner, F. An Evaluation of Artificial Fiducial Markers in Underwater Environments. OCEANS 2015, 2015, pp. 1–6. doi:10.1109/OCEANS-Genova.2015.7271491.
30. Žuži, M.; Čejka, J.; Bruno, F.; Skarlatos, D.; Liarokapis, F. Impact of Dehazing on Underwater Marker Detection for Augmented Reality. *Frontiers in Robotics and AI* **2018**, *5*, 1–13. doi:10.3389/frobt.2018.00092.
31. Andono, P.N.; Purnama, I.K.E.; Hariadi, M. Underwater image enhancement using adaptive filtering for enhanced sift-based image matching. *Journal of Theoretical and Applied Information Technology* **2013**, *51*, 392–399.
32. Ancuti, C.; Ancuti, C. Effective Contrast-Based Dehazing for Robust Image Matching. *IEEE Geoscience and Remote Sensing Letters* **2014**, *11*, 1871–1875. doi:10.1109/LGRS.2014.2312314.
33. Ancuti, C.; Ancuti, C.; Vleeschouwer, C.D.; Garcia, R. Locally Adaptive Color Correction for Underwater Image Dehazing and Matching. 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2017, pp. 997–1005. doi:10.1109/CVPRW.2017.136.
34. Gao, Y.; Li, H.; Wen, S. Restoration and Enhancement of Underwater Images Based on Bright Channel Prior. *Mathematical Problems in Engineering* **2016**, *2016*, 1–15.
35. Agrafiotis, P.; Drakonakis, G.I.; Georgopoulos, A.; Skarlatos, D. The Effect Of Underwater Imagery Radiometry On 3D Reconstruction And Orthoimagery. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* **2017**, *XLII-2/W3*, 25–31. doi:10.5194/isprs-archives-XLII-2-W3-25-2017.
36. Mangeruga, M.; Bruno, F.; Cozza, M.; Agrafiotis, P.; Skarlatos, D. Guidelines for Underwater Image Enhancement Based on Benchmarking of Different Methods. *Remote Sensing* **2018**, *10*. doi:10.3390/rs10101652.
37. Čejka, J.; Žuži, M.; Agrafiotis, P.; Skarlatos, D.; Bruno, F.; Liarokapis, F. Improving Marker-Based Tracking for Augmented Reality in Underwater Environments; Eurographics Association: Vienna, Austria, 2018; pp. 021–030. doi:10.2312/gch.20181337.
38. Pizer, S.M.; Amburn, E.P.; Austin, J.D.; Cromartie, R.; Geselowitz, A.; Greer, T.; Romeny, B.T.H.; Zimmerman, J.B. Adaptive Histogram Equalization and Its Variations. *Computer Vision Graph. Image Processing* **1987**, *39*, 355–368. doi:10.1016/S0734-189X(87)80186-X.
39. Krasula, L.; Callet, P.L.; Fliegel, K.; Klíma, M. Quality Assessment of Sharpened Images: Challenges, Methodology, and Objective Metrics. *IEEE Transactions on Image Processing* **2017**, *26*, 1496–1508. doi:10.1109/TIP.2017.2651374.
40. Limare, N.; Lisani, J.L.; Morel, J.; Petro, A.B.; Sbert, C. Simplest Color Balance. *IPOL Journal* **2011**, *1*. doi:10.5201/ipol.2011.llmps-scb.

41. Skarlatos, D.; Agrafiotis, P.; Balogh, T.; Bruno, F.; Castro, F.; Petriaggi, B.D.; Demesticha, S.; Doulamis, A.; Drap, P.; Georgopoulos, A.; Kikillos, F.; Kyriakidis, P.; Liarokapis, F.; Poullis, C.; Rizvic, S. Project iMARECULTURE: Advanced VR, iMmersive Serious Games and Augmented REality as Tools to Raise Awareness and Access to European Underwater CULTURal heritagE. Digital Heritage. Springer International Publishing, 2016, pp. 805–813.

© 2019 by the authors. Submitted to *Remote Sens.* for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).