Color Correction of Underwater Images for Aquatic Robot Inspection

Luz A. Torres-Méndez and Gregory Dudek

Centre for Intelligent Machines, McGill University,
Montreal, Quebec, H3A 2A7, CA,
latorres,dudek@cim.mcgill.ca,
WWW home page: http://www.cim.mcgill.ca/~latorres/research.html

Abstract. In this paper, we consider the problem of color restoration using statistical priors. This is applied to color recovery for underwater images, using an energy minimization formulation. Underwater images present a challenge when trying to correct the blue-green monochrome look to bring out the color we know marine life has. For aquatic robot tasks, the quality of the images is crucial and needed in real-time. Our method enhances the color of the images by using a Markov Random Field (MRF) to represent the relationship between color depleted and color images. The parameters of the MRF model are learned from the training data and then the most probable color assignment for each pixel in the given color depleted image is inferred by using belief propagation (BP). This allows the system to adapt the color restoration algorithm to the current environmental conditions and also to the task requirements. Experimental results on a variety of underwater scenes demonstrate the feasibility of our method.

1 Introduction

High quality image data is desirable for many underwater inspection and observation tasks. Particularly, vision systems for aquatic robots [3, 6, 9] must cope with a host of geometrical distortions: colour distortions, dynamic lighting conditions and suspended particles (known as 'marine snow') that are due to inherent physical properties of the marine environment. All these distortions cause poor visibility and hinder computer vision tasks, e.g., those based on stereo triangulation or on structure from motion.

Image restoration in general, involves the correction of several types of degradation in an image. Traditionally, the most common sources of degradation are due to imperfections of the sensors, or in transmission. Underwater vision is plagued by poor visibility [11, 10] (even in the cleanest water). Additional factors are the ambient light, and frequency-dependent scattering and absorption, both between the camera and the environment, and also between the light source (the sun) and the local environment (i.e. this varies with both depth and local water conditions). The light undergoes scattering along the line of sight. The result is an image that is color depleted (typically appearing bluish), blurry and out of focus. In this paper, we focus on the specific problem of restoring/enhancing the color of underwater images. The term *color* refers to the red, green and blue values (often called the color channels) for each pixel in an image. Prominent blue color of clear ocean water, apart from sky reflection, is due to selective absorption by water molecules. The quality of the water determines its filtering properties. The greater the dissolved and suspended matter, the greener (or browner) the water becomes. The time of day and cloudiness of the sky also have a great effect on the nature of the light available. Another factor is depth, once at sufficient depth, no amount of filtration can effectively restore color loss. Due to the nature of underwater optics, red light diminishes when the depth increases, thus producing blue to grey like images. By 3m in depth there is almost no red light left from the sun. By 5m, orange light is gone, by 10m most yellow is also gone. By the time one reaches 25m only blue light remains [4]. Since many (if not all) of the above factors are constantly changing, we cannot really know all the effects of water.

Color recovery is not a simple linear transform since it depends on distance and it is also affected by quantization and even light source variations. We propose a learning based Markov Random Field model for color correction based on training from examples. This allows the system to adapt the algorithm to the current environmental conditions and also to the task requirements. As proposed in[7], our approach is based on learning the statistics from training image pairs. Specifically, our MRF model learns the relationships between each of the color training images with its corresponding color depleted image. This model uses multi-scale representations of the color corrected (enhanced) and original images to construct a probabilistic enhancement algorithm that improves the observed video. This improvement is based on a combination of color matching correspondences from the training data, and local context via belief propagation (BP), all embodied in the Markov Random Field. Training images are small patches of regions of interest that capture the maximum of the intensity variations from the image to be restored.

This paper is structured as follows. Section 2 briefly consider some of the related prior work. Section 3 describes our method for color correction. Defining the MRF model and the inference approach using BP. Section 4 tests the proposed algorithm on two different scenarios with several types of experimental data each. Finally, in Section 5 we give some conclusions and future directions.

2 Related Work

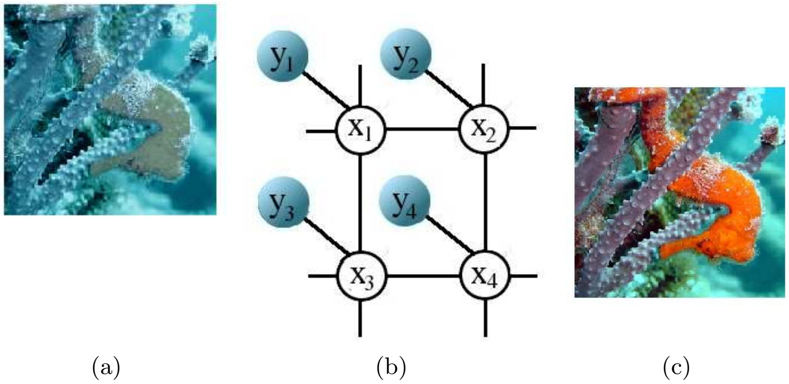
There are numerous image retouching programs on the market that have easy-to-use, semi-automated image enhancement features. But since they are directed at land-based photography, these features do not always work with underwater images. Learning to manipulate the colors in underwater images with computer editing programs requires patience. Automated methods are essential, specially for real-time applications (such as aquatic inspection). Most prior work on image enhancement tend to approximate the lighting and color processes by idealized mathematical models. Such approaches are often elegant, but may not be well suited to the particular phenomena in any specific real environment. Color restoration is an ill-posed problem since there is not enough information in the poor colored image alone to determine the original image without ambiguity. In their work, Ahlen *et al.* [1] estimate a diffuse attenuation coefficient for three wavelengths using known reflectance values of a reference gray target that is present on all tested images. To calculate new intensity values they use Beer's Law, where the depth parameter is derived from images that are taken at different depths. Additional parameters needed are the image enhancements functions built into the camera. In general, their results are good, but the method's efficiency depends highly on the previously noted parameters. In [14] a method that eliminates the backscatter effect and improves the acquisition of underwater images with very good results is presented. Their method combines a mathematical formula with a physical filter normally used for land photography. Although the method does not perform color correction, the clarity achieved on the underwater images may allow for color correction.

3 Our MRF-BP Approach for Color Correction

The solution of the color correction problem can be defined as the minimum of an energy function. The first idea on which our approach is based, is that an image can be modeled as a sample function of a stochastic process based on the Gibbs distribution, that is, as a Markov Random Field (MRF) [8]. We consider the color correction a task of assigning a color value to each pixel of the input image that best describes its surrounding structure using the training image patches. The MRF model has the ability to capture the characteristics between the training sets and then used them to learn a marginal probability distribution that is to be used on the input images. This model uses multi-scale representations of the color corrected and color depleted (bluish) images to construct a probabilistic algorithm that improves the color of underwater images. The power of our technique is evident in that only a small set of training patches is required to color correct representative examples of color depleted underwater images, even when the image contains literally no color information. Each pair of the training set is composed by a color-corrected image patch with its corresponding color-depleted image patch. Statistical relationships are learned directly from the training data, without having to consider any lighting conditions of specific nature, location or environment type that would be inappropiate to a particular underwater scene. We use a pairwise MRF model, which is of particular interest in many low-level vision problems.

3.1 The Pairwise MRF Model

Denote the input color depleted image by $\mathrm{B} = \{b_i\}, i = 1, ..., N$, where $N \in \mathbf{Z}$ is the total number of pixels in the image and b_i is a triplet containing the



(a) (b) (c)

Fig. 1. (b) Pairwise Markov Random Field used to model the joint probability distribution of the system. Observation nodes, y, represent an image patch in the bluish image (a), and hidden nodes x, an image patch in the color image (b) to be inferred.

RGB channels of pixel location i. We wish to estimate the color-corrected image $\mathrm{C} = \{c_i\}, i = 1, ..., N$, where c_i replaces the value of pixel b_i with a color value.

A pairwise MRF model (also known as *Markov network*) is defined as a set of hidden nodes x_i (white circles in the graph) representing local patches in the output image C, and the observable nodes y_i (shaded circles in the graph) representing local patches in the input bluish image B. Each local patch is centered to pixel location i of the respective images. Figure 1 shows the MRF model for color correction.

Denoting the pairwise potentials between variables x_i and x_j by ψ_{ij} and the local evidence potentials associated with variables x_i and y_i by ϕ_i (see Figure 2), the joint probability of the MRF model under variable instantiation $\mathrm{x} = (x_1, ..., x_N)$ and $\mathrm{y} = (y_1, ..., y_N)$, can be written [2, 8] as:

$$P(\mathrm{x}, \mathrm{y}) = \frac{1}{Z} \prod_{ij} \psi_{ij}(x_i, x_j) \prod_{i} \phi_i(x_i, y_i), \tag{1}$$

where Z is the normalization constant. We wish to maximize $P(\mathrm{x}, \mathrm{y})$, that is, we want to find the most likely state for all hidden nodes x_i, given all the evidence nodes y_i.

The compatibility functions allows to set high (or low) compatibilities to neighboring pixels according to the particular application. In our case, we wish to preserve discontinuities (edges) in the input (color depleted) image to avoid over smoothing the color corrected image. Thus, we set high compatibility between neighboring pixels that have similar colors, and low compatibility between neighboring pixels with abrupt change in color values. These potentials are used in messages that are propagated between the pixels to indicate what color or combination of intensities each image pixel should have.

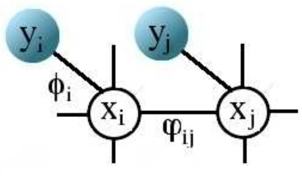


Fig. 2. The potential functions ϕ and ψ define the compatibilities between nodes in the Markov network.

A color pixel value in C is synthesized by estimating the maximum a posteriori (MAP) solution of the MRF model using the training set. The MAP solution of the MRF model is:

$$\mathbf{x}_{MAP} = \arg \max_{\mathbf{x}} P(\mathbf{x} \mid \mathbf{y}), \tag{2}$$

where

$$P(\mathbf{x} \mid \mathbf{y}) \propto P(\mathbf{y} \mid \mathbf{x}) P(\mathbf{x}) \propto \prod_i \phi_i(x_i, y_i) \prod_{(i,j)} \psi_{ij}(x_i, x_j) \tag{3}$$

Calculating the conditional probabilities in an explicit form to infer the exact MAP in MRF models is intractable. We cannot efficiently represent or determine all the possible combinations between pixels with its associated neighborhoods. Various techniques exist for approximating the MAP estimate, such as Markov Chain Monte Carlo (MCMC), iterated conditional modes (ICM), maximizer of posterior marginals (MPM), etc. See [5] for a comparison. In this work, we compute a MAP estimate, by using a learning-based framework on pairwise MRFs, as proposed by [7], using belief propagation (BP).

The compatibility functions $\phi(x_i, y_i)$ and $\psi(x_i, x_j)$ are learned from the training set using the patch-based method in [7]. They are usually assumed to obey a Gaussian distribution to model Gaussian noise. The $\phi_i(x_i, y_i)$ compatibility function is defined as follows

$$\phi_i(x_i, y_i) = \mathrm{e}^{-|y_i - y_{x_i}|^2 / 2\sigma_i^2} \tag{4}$$

where x_i is a color-corrected patch candidate, y_{x_i} is the corresponding bluish patch of x_i, and y_i is the bluish patch in the input image.

The image is divided so that the corresponding color-corrected patches overlap. If the overlapping pixels of two node states match, the compatibility between those states is high. We define $\psi(x_i, x_j)$ as:

$$\psi_{ij}(x_i, x_j) = \mathrm{e}^{-d_{ij}(x_i, x_j) / 2\sigma_i^2} \tag{5}$$

where d_{ij} is the difference between neighborhoods i and j (Section 3.3 defines the precise similarity measure we use).

Images in the training set are pairs of small image regions of the bluish image with its corresponding color-corrected image, thus the compatibility functions depend on each particular input image.

3.2 MRF-MAP inference using BP

Belief propagation (BP) was originally introduced as an exact algorithm for tree-structured models [12], but it can also be applied for graphs with loops, in which case it becomes an approximate algorithm, leading often to good approximate and tractable solutions [15]. For MRFs, BP is an inference method to efficiently estimate Bayesian beliefs in the network by the way of iteratively passing messages between neighboring nodes.

The message send from node i to any of its adjacent nodes $j \in \mathrm{N}(i)$ is

$$m_{ij}(x_j) = Z \sum_{x_i} \psi(x_i, x_j) \phi(x_i, y_i) \prod_{k \in \mathrm{N}(i) \backslash \{j\}} m_{ki}(x_i) \tag{6}$$

where Z is the normalization constant. The maximum a posteriori scene patch for node i is:

$$x_{iMAP} = \arg \max_{\mathbf{x}_i} \phi(x_i, y_i) \prod_{j \in \mathrm{N}(i)} m_{ji}(x_i). \tag{7}$$

The BP algorithm is not guaranteed to converge, but if it does so, then it converges to a local stationary point of the Bethe approximation to the free energy [17]. In our experiments, the BP algorithm usually converges in less than 10 iterations. And it is also notable that BP is faster than many traditional inference methods.

Candidate states for each patch are taken from the training set. Fore each bluish patch in the image, we search the training set for patches that best resemble the input. The color-corrected patches corresponding the best k patches are used as possible states for the hidden nodes.

The algorithm for color correction can be summarized as follows:

1. Divide the training images (both the bluish and color images) into small patches, which form the sets of x_i's and y_i's.
2. For each input patch y_i, find the k closest y_{x_i}'s. The corresponding x_i's are the candidates for that patch. Calculate the compatibility function $\phi(x_i, y_i)$ according to Eq. 4.
3. For each pair of neighboring input patches, calculate the $k \times k$ compatibility function $\psi(x_i, x_j)$ according to Eq. 5.
4. Estimate the MRF-MAP solution using BP.
5. Assign the color value of the center pixel of each estimated maximum probability patch x_{iMAP} to the corresponding pixel in output image C.

3.3 Implementation issues

Measuring the dissimilarity between image patches is of crucial for obtaining quality results, especially when there is a prominent color (blue or green) as in underwater images. Color information can be specified, created and visualized by different color spaces (see [16] for more information about color spaces). For example, the *RGB* color space, can be visualized as a cube with red, green and blue axes. Color distance is a metric of proximity between colors (e.g. Euclidean distance) measured in a color space. However, color distance does not necessarily correlate with *perceived* color similarity. Different applications have different needs which can be handled better using different color spaces. For our needs it is important to be able to measure differences between colors in a way that matches perceptual similarity as good as possible. This task is simplified by the use of *perceptually uniform* color spaces. A color space is perceptually uniform if a small change of a color will produce the same change in perception anywhere in the color space. Neither RGB, HLS or CIE XYZ is perceptually uniform.

The (nonlinear) conversions from RGB to CIE Lab are given by: [1]

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$\mathrm{L}^* = \begin{cases} 116(Y/Y_n)^{1/3} - 16 & \text{if } Y/Y_n > 0.008856 \\ 903.3(Y/Y_n) & \text{otherwise} \end{cases}$$

$$a^* = 500[f(X/X_n)^{1/3} - f(Y/Y_n)^{1/3}]$$

$$b^* = 200[f(Y/Y_n)^{1/3} - f(Z/Z_n)^{1/3}]$$

where

$$\mathrm{f(t)} = \begin{cases} t^{1/3} & \text{if } Y/Y_n > 0.008856 \\ 7.787t + 16/116 & \text{otherwise} \end{cases}$$

We use the CIE *Lab* space which was designed such that the equal distances in the color space represent equal perceived differences in appearance. Color difference is defined as the Euclidean distance between two colors in this color space:

$$\Delta E^*_{ab} = \sqrt{(\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2} \qquad (8)$$

where ΔL^*, Δa^*, and Δb^* are the differences between two color pixel values.

This is the similarity measure used to select possible candidates to define the compatibility functions and also to evaluate the performance of our method. Our algorithm uses a pixel-based synthesis, i.e. one pixel (color) value c_i is estimated at a time.

[1] Following ITU-R Recommendation BT.709, we use D_{65} as the reference white point so that $[X_n, Y_n, Z_n] = [0.9504511.088754]$ (see [13])

4 Experimental results

We test the proposed approach in two different scenarios. In the first scenario, we use color underwater images available on the web [2] as our ground truth data. These images were taken with a professional camera and in most of the cases they were also enhanced by using a commercial software. The second scenario, involves the acquisition of underwater video by our aquatic robot. Sections 4.1 and 4.2 describe these scenarios with the experimental results.

4.1 Scenario 1

In order to simulate the effects of water, an attenuation filter were applied to each of the color underwater image. Figure 3a shows the ground truth (color) image and Figure 3b, the simulated (color depleted) image after applying the attenuation filter. Since we have ground truth information, we can compute the performance of our algorithm. The images in the training set correspond to small image regions extracted from the ground truth image and the color depleted image (see Figure 4).

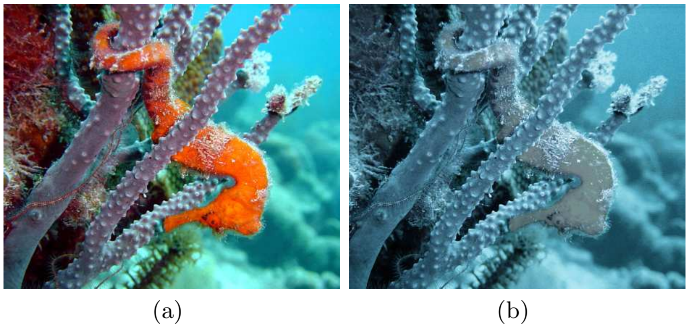


Fig. 3. (a) The ground truth (color) image. (b) The simulated bluish image (this is the test image to be color corrected by our algorithm).

These images correspond to regions of interest in terms of the variations in pixel color values , thus the intention is that they capture the intrinsic statistical dependencies between the color depleted and ground truth pixel values. The size of the neighborhoods in all experiments were 5×5 pixels, the overlapping area between image patches 2×5 pixels, and the number of possible candidates k, was fixed to be 10. Figure 5a shows the training image patches from where our algorithm learns the compatibility functions and Figure 5b shows the resulted image after running our learning-based method. The color-corrected image *looks*

[2] http://www.pbase.com/imagine

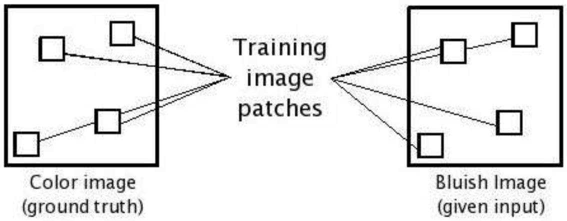
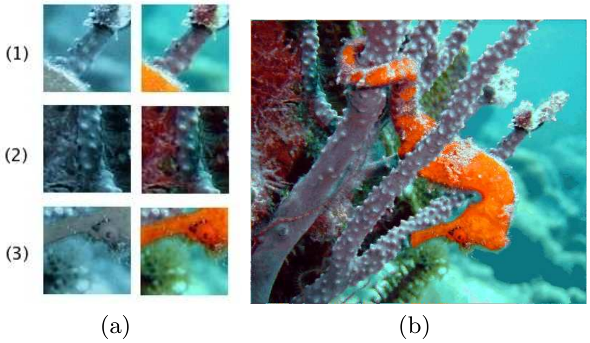


Fig. 4. Diagram showing how the training image pairs are acquired for the Scenario 1.



(a) (b)

Fig. 5. (a) The training image patches used to learn the compatibility functions. (b) The color corrected image.

good, the discontinuities and edges are preserved since our method assign colors pixel by pixel, thus avoiding over-smoothing. Also, there are no sudden changes in color which are typically both unrealistic and perceptually unappealing. To evaluate the performance of our algorithm, we compute the mean absolute residual (MAR) error between the ground truth and the color corrected images. As mentioned in Section 3.3, the CIELab metric was used to calculate the similarities between pixels in the images. For this case, the MAR error is 6.5. For comparison purposes, we calculate the MAR error between the input (color depleted) image and the ground truth image, this is 22.03.

Using the same input image (Figure 5b), we now show how the final result varies depending on the training data. In Figure 6, 4 examples when using different training pairs are shown. For example, Figure 6a shows a color-corrected image when using training pairs (1) and (3) (see Figure 5a). The MAR errors are 9.43, 9.65, 9.82, and 12.20, respectively. It can be seen that the resulting images are limited to the statistical dependencies captured by the training pairs.

Three more examples of underwater scenes are shown in Figure 7. Each row shows from left to right, the ground truth color image, the input bluish image

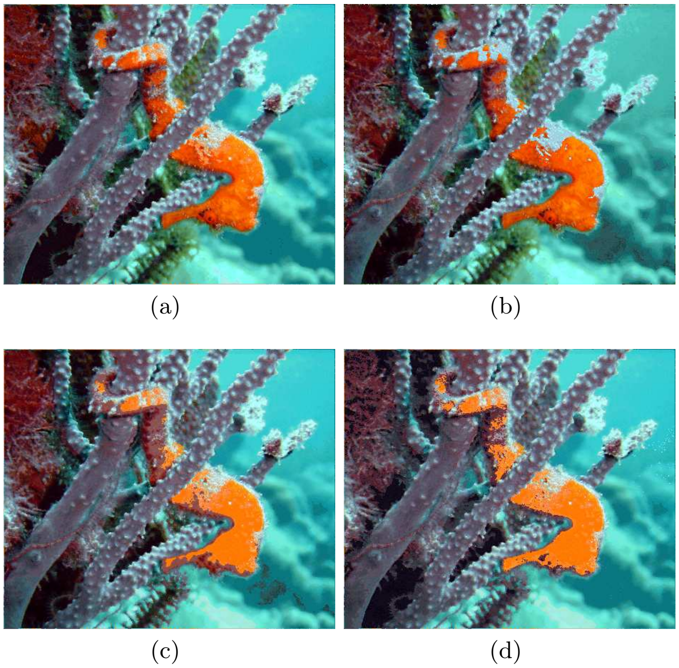


Fig. 6. Color correction results using different training sets. The input image is shown in Figure 3b. The training pairs (labeled) are shown in Figure 5a. Results using training pair (a) (1) and (3); (b) (2) and (3); (c) (1) and (2), and (d) (1).

and the color corrected image after running our algorithm. The training image regions are shown by squares in the corresponding color and bluish images. In general the results looks very good. For the last two examples, the size of the image patches in the training set is very small and enough to capture all the statistical dependencies between bluish and color information, as a result, the number of total comparisons in our algorithm is reduced and speed is achieved.

It was previously mentioned, that underwater images also contain some blurriness. In Figure 8, we show an example of applying our algorithm to a blurry and color depleted image at the same time. From left to right are, the ground truth image, the input image given to our algorithm and the color-corrected and deblurred image after running our algorithm.

4.2 Scenario 2: The aquatic robot in action

As our aquatic robot [9] swims through the ocean, it takes video images. Figure 9 shows a picture of our aquatic robot in action.

In order to be able to correct the color of the images, training data from the environment that the robot is currently seeing needs to be gathered. How can

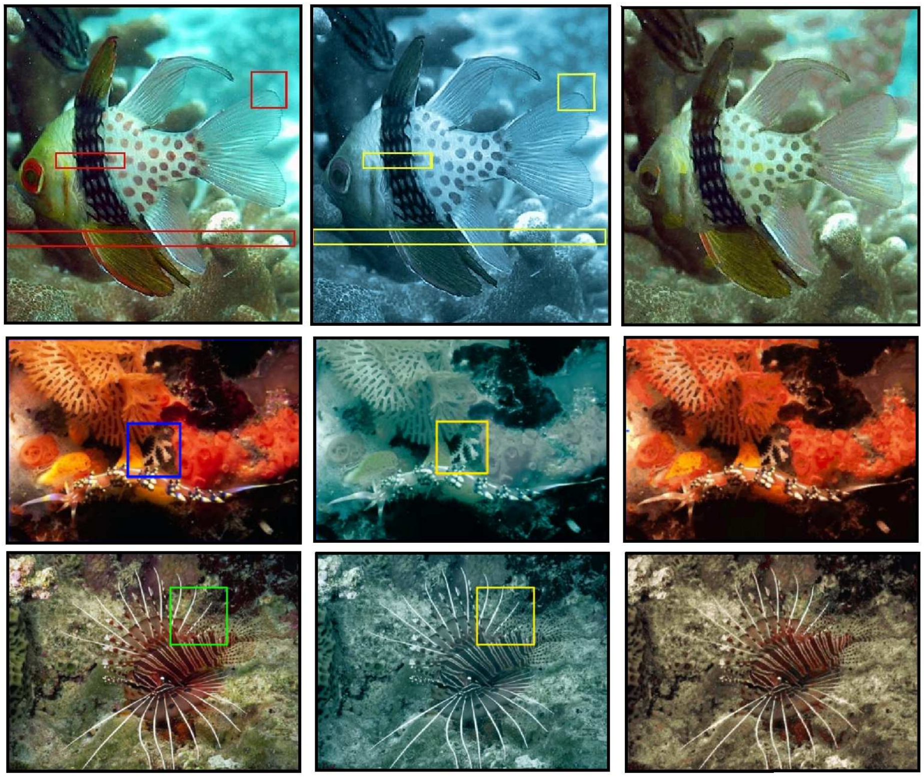


Fig. 7. More examples. The training pairs are indicated by the squares in the original and input images respectively.

better images be acquired? As light is absorbed selectively by water, not only does it get darker as you go deeper, but there is a marked shift in the light source color. In addition, there are non-uniformities in the source amplitude. Therefore, the aquatic robot needs to bring its own source of white light on it. However, due to power consumption, the light cannot be left turned on. Therefore, only at certain time intervals, the robot stops, turns its light on and take an image. These images are certainly much better, in terms of color and clarity, than the previous ones, and they can be used to train our algorithm to color correct neighboring frames (under the assumption that neighboring frames are similar). Figure 10 shows this scenario, here frame t_3 represents the image pair to be used to train our model for color correction.

Now we show an example. Figures 11a,b show the training image pair captured at time t. The robot moves around and then at time $t + \delta$ takes an image (Figure 11c), which is input to our algorithm. The resulting color-corrected image is shown in Figure 11d. Since we do not have ground truth data for this

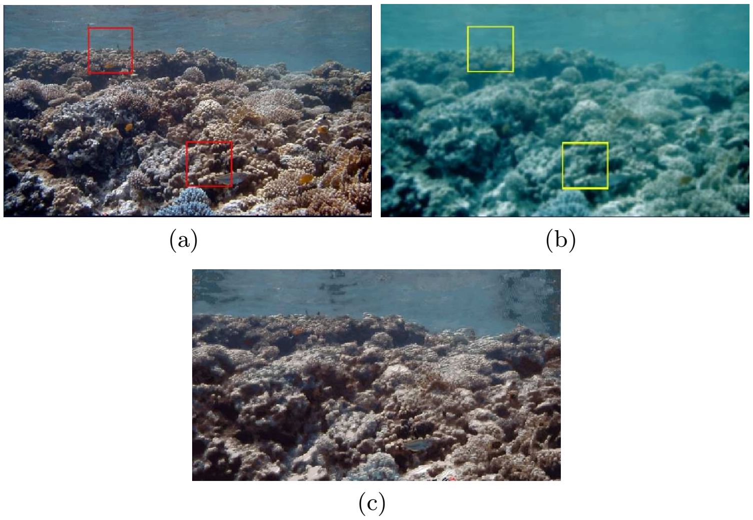

(a) (b) (c)

Fig. 8. An example of color correcting and deblurring at the same time. The training pairs are indicated by the boxes in the original (a) and input images (b) respectively. (c) is the color-corrected and deblurred image.

scenario, we cannot measure the performance of our algorithm, however it can be seen that the resulting image looks visually good.

5 Summary and Conclusions

Color restoration and image enhancement are ubiquitous problems. In particular, underwater images contain distortions that arise from multiple factors making them difficult to correct using simple methods. In this paper, we show how to formulate color recovery and more general enhancement as an energy minimization problem using learned constraints. This approach's novelty lies in using a pair of images to constrain the reconstruction. There are some factors that influence the quality of the results, such as the adequate amount of reliable information as an input and the statistical consistency of the images in the training set.

Acknowledgments

We would like to thank photographer Ellen Muller for allowing the use of her underwater images for research purposes.

References

1. J. Åhlén and D. Sundgren. Bottom reflectance influence on a color correction algorithm for underwater images. In *Proc. SCIA*, pages 922–926, 2003.

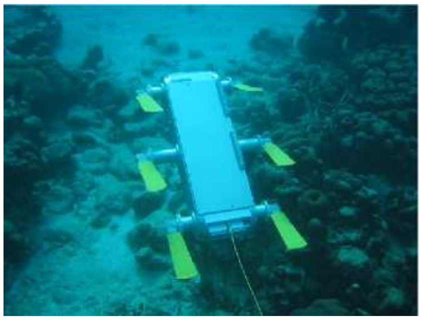


Fig. 9. The aquatic robot.

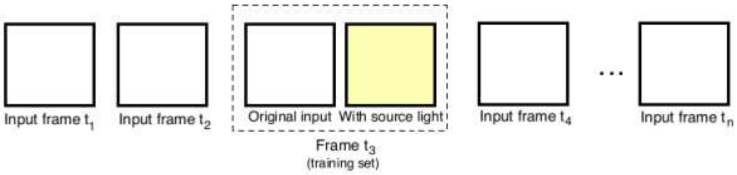


Fig. 10. The scenario 2.

2. J. Besag. Spatial interaction and the statistical analysis of lattice systems (with discussion). *J. Royal Statistics Society B*, 36:192–326, 1974.
3. T. Boult. DOVE: Dolphin omni-directional video equipment. In *Proc.Int. Conf. Robotics and Automation*, pages 214–220, 2000.
4. J. Dera. *Marine Physics*. Elsevier, 1992.
5. R.C. Dubes, A.K. Jain, S.G. Nadabar, and C.C. Chen. Mrf model-based algorithms for image segmentation. In *Proc. of IEEE Intl. Conf. on Pattern Recognition (ICPR)*, pages 808–814, 1990.
6. G.L. Foresti. Visual inspection of sea bottom structures by an autonomous underwater vehicle. *IEEE Trans. Syst. Man and Cyber., Part B*, 31:691–795, 2001.
7. W.T. Freeman, E.C. Pasztor, and O.T. Carmichael. Learning low-level vision. *International Journal of Computer Vision*, 20(1):25–47, 2000.
8. S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.
9. C. Georgiades, A. German, A. Hogue, H. Liu, C. Prahacs, A. Ripsman, R. Sim, L. A. Torres-Méndez, P. Zhang, M. Buehler, G. Dudek, M. Jenkin, and E. Milios. AQUA: an aquatic walking robot. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 3525–3531, Sendai, Japan, September 2004. IEEE/RSJ, IEEE Press.
10. S. Harsdorf, R. Reuter, and S. Tonebon. Contrast-enhanced optical imaging of submersible targets. In *Proc. SPIE*, volume 3821, pages 378–383, 1999.
11. J.S. Jaffe. Computer modeling and the design of optimal underwater imaging systems. *IEEE J. Oceanic Engineering*, 15:101–111, 1990.
12. J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, Inc., 1988.

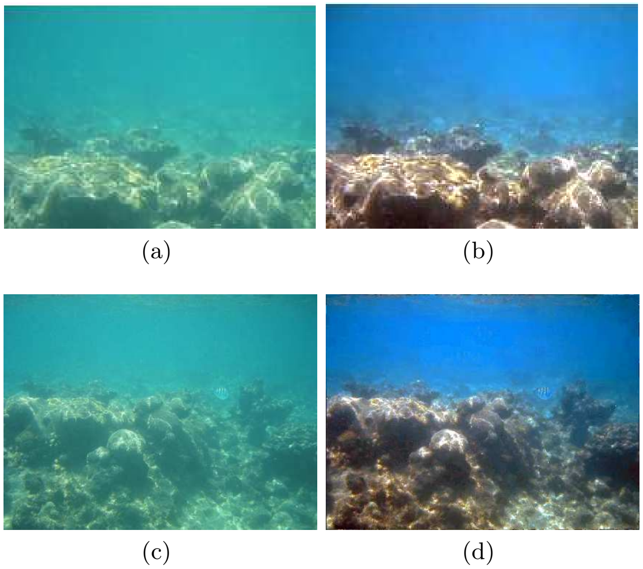


Fig. 11. (a)-(b) The training image pair captured at frame t. (c) Image taken at frame $t + \delta$ and input to our algorithm. (d) The color corrected image.

13. C. Poynton. *A Technical Introduction to Digital Video*. Wiley, NewYork, 1996.
14. Y.Y. Schechner and N. Karpel. Clear underwater vision. In *Proc. of Intt. Conference of Vision and Pattern Recognition*, volume 1, pages 536–543, 2004.
15. Y. Weiss. Belief propagation and revision in networks with loops. Technical report, Berkeley Computer Science Dept., 1998.
16. G. Wyszecki and W.S. Stiles. *Color Science: Concepts and Methods, Quantitative Data and Formulae*. Wiley, NewYork, 1982.
17. J. Yedidia, W. Freeman, and Y. Weiss. Constructing free energy approximations and generalized belief propagation algorithms. Technical report, Mitsubishi Electrical Research Laboratories, Inc., 2004.