Feature Matching In Underwater Environments Using Sparse Linear Combinations

Kenton Oliver
Department of Computer Science and Engineering
University of South Carolina
oliverwk@cec.sc.edu

Weilin Hou
Naval Research Lab, Code 7333
weilin.hou@nrlssc.navy.mil

Song Wang
Department of Computer Science and Engineering
University of South Carolina
songwang@cec.sc.edu

Abstract

Feature matching is a key, underlying component in many approaches to object detection, localization, and recognition. In many cases, feature matching is accomplished by nearest neighbor methods on extracted feature descriptors. This methodology works well for clean, out-of-water images; however, when imaging underwater, even an image of the same object can be drastically different due to varying water conditions. As a result, descriptors of the same point on an object may be completely different between the clean and underwater images, and between different underwater images taken under varying imaging conditions. This makes feature matching between such images a very challenging problem. In this paper, we present a new method for feature matching by first synthetically constructing a feature codebook for all template features by simulating different underwater imaging conditions. We then approximate the target feature by a sparse linear combination of the features in the constructed codebook. The optimal sparse linear combination is found by compressive sensing algorithms. In the experiments, we show that the proposed method can produce better feature matching performance than the nearest neighbor approach and associated naïve extensions.

1. Introduction

Detection, description, and matching of discriminative feature points from an image are fundamental problems in computer vision and have been studied for many years. Many feature detectors, descriptors, and feature matching algorithms have been developed and play key roles in many vision applications, such as image stitching [1, 10], image registration [6, 16], object detection [20], object localization [12], and object recognition [14]. In practice, we usually require feature descriptors to be invariant to certain image spatial transformations, such as scaling and rotation.

Geodesic Invariant Histograms (GIH) [13] model a grayscale image as a 2-dimensional surface embedded in 3-dimensional space, where the height of the surface is defined by the image intensity at the corresponding pixel. Under this surface model a feature descriptor, based on geodesic distances on the surface, is then defined which is invariant to some general image deformations. A local-to-global framework was adopted in [3] where multiple support regions are used for describing the feature at a single point. This removes the burden of finding the optimal scale and both local and global information is embedded in its descriptor. The Scale-Invariant Feature Transform (SIFT) [15] is a well-known choice for detecting and describing features. Comparison studies [17] have shown that SIFT and its derivatives [11, 19, 17] perform better than other feature detectors in various tasks. SIFT is rotation and scaling invariant and has been shown to be invariant to small changes in illumination and perspectives up to 50 degrees.

All of the previously mentioned feature detectors and descriptors only address invariance in the spatial domain. They are not invariant when the considered image undergoes a destructive intensity transformation, which changes the image intensity values substantially, inconsistently, and irreversibly. Such transformations often significantly increase the complexity in discerning any underlying features and structures in the image, as shown in [17]. A typical example that may be encountered is intensity transformation introduced by underwater imaging. Light behaves differ-

ently underwater [18], and when dealing with impure water, issues such as turbulence, air bubbles, and particles such as sediments and organic matter can absorb and scatter light, resulting in a very blurry and noisy image. Since available feature descriptors are not invariant under such intensity transformations, matching the features detected from an underwater image and a clean out of water image, or the features detected from two underwater images taken at different underwater conditions, is a very challenging problem, as illustrated in Fig. 1.

In this paper, we present a new method for feature matching by first synthetically constructing a feature codebook for all template features by simulating different underwater imaging conditions. We then approximate the feature to be matched by a sparse linear combination of the features in the constructed codebook. The optimal sparse linear combination is found by the compressive sensing algorithm. The template feature in the codebook that contributes most to the resulting sparse linear combination is selected as the matched feature. In the experiments, we show that the proposed method can produce better feature matching than the widely used nearest neighbor approach and its associated naïve extensions.
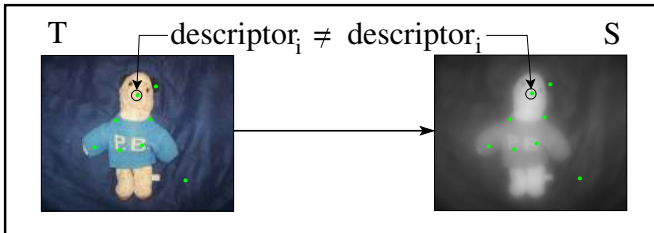

Figure 1. Illustration of the same feature with very different descriptors due to varying underwater conditions.

The remainder of this paper is organized as follows: Section 2 gives a brief introduction to the difficulties of imaging underwater and the models we use to simulate this, Section 3 covers SIFT matching and introduces a simple extension to SIFT and an explanation of the proposed method, Section 4 presents the experimental setup and the performance results, and Section 5 presents the problems that still need to be addressed for this approach to be more feasible; this is followed by a conclusion.

2. Imaging In Underwater Environments

Underwater Imaging is an area with many applications including defense, mine countermeasures, security, search and rescue, and conducting scientific experiments in harsh, unreachable environments. On a clear day, a person can see miles to the horizon out-of-water, and yet in many underwater conditions, one cannot see more than a few meters; and what can be seen is blurred and difficult to discern. This reduction in visibility is due to the absorption and scattering

of light by the water and particles in the water. There are numerous particles such as sediment, plankton, and organic cells in the water which cause this scattering and absorption. Even water turbulence and bubbles effect how light is transmitted. Fig. 2 illustrates how light can be scattered by a particle. Light that is spread out by this scattering is what causes the blurriness and fuzziness common in underwater images (see Fig. 3).
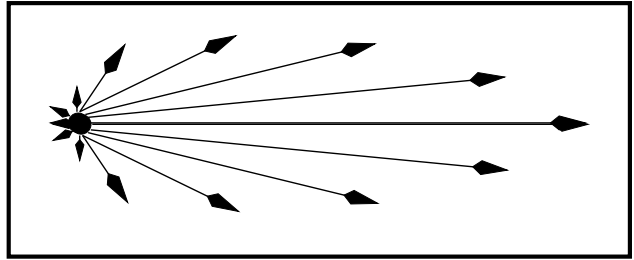

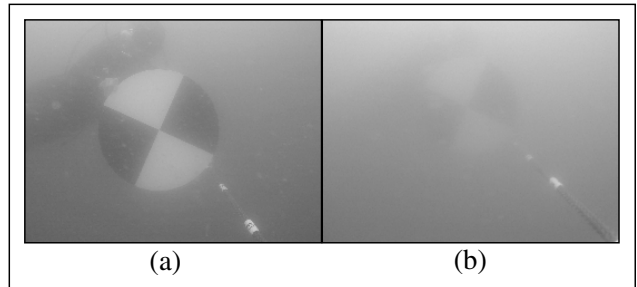Figure 2. How a particle might scatter light in the water.

(a)   (b)

Figure 3. underwater images of a Secchi disk, an instrument used to measure diver visibility. (a) Diver with Secchi disk: notice bubbles and general fuzziness. (b) Same diver and Secchi disk taken from only a few meters further away from (a). Notice how this small change in distance greatly reduced the visibility of the disk.

This absorption and scattering of light in water can be modeled mathematically, and much work has been done to develop robust models to this effect [9, 4, 8, 7]. These models are typically some form of a *point spread function* (PSF) which models a system's response to an impulse signal (point source). For this work, we use a simplified version of Dolin's PSF model [4, 8], to simulate underwater conditions. Given an out-of-water image, convolution with the PSF creates a synthetic underwater image. Dolin's model takes the form

$$G(\theta_b, \tau_b) = \frac{\delta(\theta_q)}{\pi \theta_q} \exp(-\tau_b) + 0.525$$
$$\cdot \frac{\tau_b}{\theta_q} \exp(-2.6\theta_q^{0.7} - \tau_b) + \frac{\beta_2^2}{2\pi}[2 - (1 + \tau_b)\exp(-\tau_b)]$$
$$\cdot \exp[-\beta_1(\beta_2\theta_q)^{\frac{1}{3}} - (\beta_2\theta_q)^2 + \beta_3]$$

where, θ_q is the scattering angle—the angle at which light is reflected away from its original direction—and $\tau_b = \tau\omega$, where τ is the optical depth and ω is the single scattering

albedo, the ratio of light scattered to total light attenuation. For the inclined reader more details can be found in [4, 8]. In this paper we will use the notation $\mathrm{PSF}(\cdot, \tau, \omega)$ to refer to the operation of convolution with a PSF with parameters τ and ω.

3. Feature Matching

Given two images of an object, S and T, imaged under different conditions (i.e. out-of-water, in clean water, in slightly turbid water, etc), we would like to determine if these two images contain the same (or similar) object(s). One approach is as follows:

1. Detect sets of features from each image S and T.

2. Match features between S and T.

3. "Goodness" of the matching gives likelihood of being the same object.

As stated previously, available detectors are not designed to account for the large differences in descriptors that are present due to the underwater conditions. This makes the matching step difficult because the descriptors lose a great deal of their discriminative power due to these conditions. We would like to investigate the problem of matching under these conditions. In the rest of this section we examine three matching algorithms to address these issues. First we look at how SIFT points are matched, then give a simple extension for SIFT matching which can address some of these concerns, and finally present the proposed matching approach.

3.1. General Feature Matching

In general feature matching is based on a comparison of Euclidean distances between feature descriptors. Some schemes match if this distance is below a given threshold, Let P and Q be two feature points with descriptors $\mathbf{p}$ and $\mathbf{q}$ respectively. Matching with a threshold is defined as

$$\mathrm{match_t}(P,Q) = \begin{cases} 1 & \|\mathbf{p}-\mathbf{q}\|_2 \leq t \\ 0 & \|\mathbf{p}-\mathbf{q}\|_2 > t. \end{cases} \quad (1)$$

This method of matching has the added difficulty that a good threshold must be chosen. Another choice would be to match nearest neighbors: a descriptor $\mathbf{p}$ is matched to $\mathbf{q}$ if it is closer to $\mathbf{q}$ than any other descriptor. Let $P^1, \ldots, P^N$ be a set of feature points with respective descriptors $\mathbf{p}^1, \ldots, \mathbf{p}^N$. Then nearest neighbor matching is defined as

$$\mathrm{match_{NN}}(P^k,Q) = \begin{cases} 1 & k = \underset{i=1,\ldots,N}{\mathrm{argmin}} \|\mathbf{p}^i-\mathbf{q}\|_2 \\ 0 & k \neq \underset{i=1,\ldots,N}{\mathrm{argmin}} \|\mathbf{p}^i-\mathbf{q}\|_2 \end{cases} \quad (2)$$

To match SIFT points, an approximate nearest neighbor approach is implemented. An approximation is used because efficient nearest neighbor algorithms usually do not perform better than brute force search in dimensions larger than 10. SIFT descriptors are of dimension 128, so using k-d trees to do nearest neighbor search is not very efficient. SIFT descriptors are matched with the Best Bin First algorithm, which is a nearest neighbor approximation which returns the nearest neighbor with high probability [15].

3.2. A Simple Extension

This simple extension is derived from the intuition that nearest neighbor matching will fail if the descriptors are highly varied due to very different water conditions; however, if the water conditions show some similarity then the difference in descriptor will not be as great and a nearest neighbor approach should handle this reasonably well. With this in mind, we can process two images, S and T, taken in very different water conditions, by simulating another underwater environment on S, denoted $\mathrm{PSF}(S, \tau, \omega)$, such that this environment is closer to that of Ts. We can then easily apply a nearest neighbor approach to match them.

There are a few complications with this approach. First, it is unclear how, from T, can we determine the correct parameters, τ and ω, needed to estimate its conditions. Second, if S was also taken underwater, applying a PSF to an underwater image may not correctly model the underwater environment. To address these issues our approach does not try to estimate the exact parameters for the PSF, instead sampling a range of parameters and attempting to match with each one, choosing the matching with the best matching score. See Fig. 4 for an example.

Formally, let $S_1, \ldots, S_n$ be the set of images obtained from simulating n different PSFs on S: $S_i = \mathrm{PSF}_i(S) = \mathrm{PSF}(S, \tau_i, \omega_i)$. The notation PSF_i is used as shorthand to refer to the PSF with parameters τ_i and ω_i. Feature points are detected on the original S and then the points are used to build descriptors on each S_i. So for any feature point $P^j \in S$, $j = 1, \ldots, N$, there is a corresponding feature point $P_i^j \in S_i$ with descriptor $\mathbf{p}_i^j$. With this notation, superscripts denote a particular feature point and subscripts denote the simulated condition under which the descriptor was obtained. The extension can then be formulated as

$$\mathrm{match_{ENN}}(P^k,Q) =$$
$$\begin{cases} 1 & k = \underset{j=1,\ldots,N}{\mathrm{argmin}} [\underset{i=1,\ldots,n}{\min} \|\mathbf{p}_i^j-\mathbf{q}\|_2] \\ 0 & k \neq \underset{j=1,\ldots,N}{\mathrm{argmin}} [\underset{i=1,\ldots,n}{\min} \|\mathbf{p}_i^j-\mathbf{q}\|_2] \end{cases} \quad (3)$$

If we can obtain a set of PSFs that closely approximate the conditions in T, then this approach should work well. Because the parameter space is continuous, it is infeasible to compute all possible PSFs, but the parameter space can
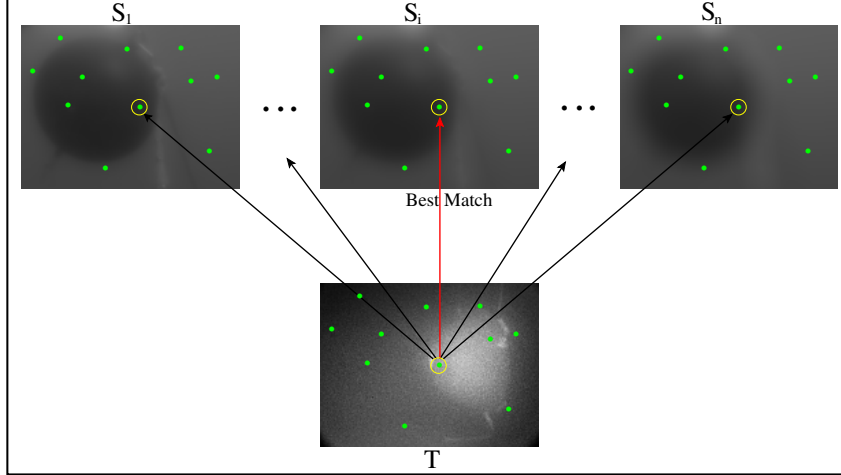
Figure 4. Illustration of matching with the best guess. Each S_i is the original S convoluted with a PSF with parameters τ_i and ω_i. Matching is carried out for each S_i and the matching with best matching score is chosen.

be discretized while still representing much of the variation in the PSF. Because convolution is not a cheap operation, trying all possible PSFs in our discretized parameter space is inefficient, so we would like to use as few PSFs as possible. Lastly, this approach will suffer if noise or conditions not represented by the PSF model are present.

3.3. Compressive Sensing Approach

The proposed matching algorithm is motivated by the previous extension of the nearest neighbor approach: Even the best simulated environment might not be a very good approximation, so can the true conditions be better simulated by finding the best combination of simulated environments that explains the observed feature descriptors well?

One example where this might be the case is to consider the image of a torpedo taken from the front looking down its length, so you can see the side of the torpedo and its nose is much closer than its tail. Our model of PSF assumes a single optical depth τ, but in this case the nose of the torpedo obviously has a different optical depth than the tail and points in between. Our chosen PSF does not consider this, but it can easily occur in real situations. Even if we use a PSF which represents the conditions at the nose and another which represents the conditions at the tail, the points in between might not be well represented. But if we assume that the PSF changes fairly smoothly from nose to tail, then some linear combination of the PSFs could potentially capture the conditions in the middle.

As before, we have $S_i = \mathrm{PSF}_i(S)$, with feature points $P^1, \ldots, P^N$ and corresponding feature descriptors $\mathbf{p}_i^1, \ldots, \mathbf{p}_i^N$, all for $i = 1, \ldots, n$. Then we want to find a combination of descriptors such that

$$\mathbf{q} = \sum_{i=1}^{n} \beta_i \mathbf{p}_i^k.$$

for some k (see Fig. 5). A sharp equality is probably unlikely, so in practice we want the combination that can best approximate $\mathbf{q}$.

To facilitate this we build a feature descriptor codebook containing all of the simulated descriptors. The codebook is a matrix Φ with each column a $\mathbf{p}_i^j$: $\Phi = [\mathbf{p}_1^1 \mathbf{p}_2^1 ... \mathbf{p}_n^1 ... \mathbf{p}_n^N]$. Then, to best approximate $\mathbf{q}$, a coefficient vector $\boldsymbol{\alpha}$ is needed such that

$$\Phi \boldsymbol{\alpha} = \mathbf{q}. \qquad (4)$$

It should now be noted that this coefficient vector $\boldsymbol{\alpha}$ should, ideally, be sparse or have most values close to zero. To see why, assume that the true matching for feature Q is P^k, Then only the coefficients corresponding to $\mathbf{p}_1^k, \ldots, \mathbf{p}_n^k$ should be large, or non-zero, all others should be zero or close to zero. To enforce this we find $\boldsymbol{\alpha}$ which is sparsest

$$\hat{\boldsymbol{\alpha}} = \underset{\Phi \boldsymbol{\alpha} = \mathbf{q}}{\mathrm{argmin}} \|\boldsymbol{\alpha}\|_0. \qquad (5)$$

Solving this problem, (Eq. 5) is well-known to be NP-hard; however, since our solution is sparse this can be solved with good approximation by L-1 minimization according to compressed sensing [2, 5]

$$\hat{\boldsymbol{\alpha}} = \underset{\Phi \boldsymbol{\alpha} = \mathbf{q}}{\mathrm{argmin}} \|\boldsymbol{\alpha}\|_1. \qquad (6)$$

Now being able to solve for sparse linear combinations of our codebook, we compare how closely each set of descriptors, $\mathbf{p}_1^j, \ldots, \mathbf{p}_n^j$, for each $j = 1, \ldots, N$, approximates $\mathbf{q}$. Then P^k is matched to Q if it has the smallest residual,

$$\mathrm{match}_{\mathrm{CS}}(P^k, Q) = \begin{cases} 1 & k = \underset{j=1,\ldots,N}{\mathrm{argmin}} \|\mathbf{q} - \sum_{i=1}^{n} \alpha_i^j \mathbf{p}_i^j\|_2 \\ 0 & k \neq \underset{j=1,\ldots,N}{\mathrm{argmin}} \|\mathbf{q} - \sum_{i=1}^{n} \alpha_i^j \mathbf{p}_i^j\|_2 \end{cases} \qquad (7)$$
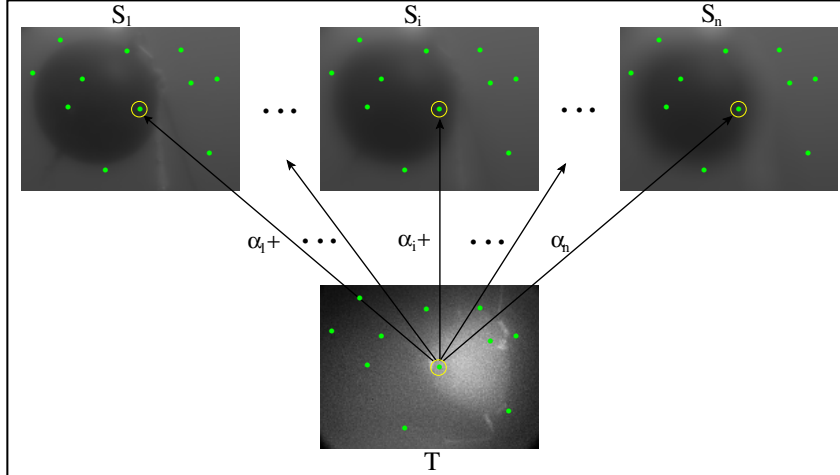
Figure 5. Matching using a linear combination of descriptors to estimate the matching descriptor.

This formulation of the matching problem allows us to perform matching that can account for variation under the water by matching against a sparse linear combination of descriptors as opposed to a single descriptor. This combination of descriptors is shown to give more discriminative matching power than the previously detailed approaches.

4. Experiments and Results

4.1. Experiment Setup

To quantitatively evaluate the feature matching, we need a ground-truth matching to compare against. In this paper, we construct synthetic underwater images with known ground-truth feature matching for performance evaluation. Underwater images usually contain relatively few objects on dark backgrounds. We pick several such clean images taken out of water as the base for synthesizing test underwater images. One of these images (shown in Fig. 6) is a stuffed bear chosen for the textured fur and the other is a Secchi disk, an instrument used to measure underwater visibility. In each of our experiments, we run all three matching algorithms described in Section 3. In addition, we set $n = 2$ to construct the feature codebook. A smaller-size codebook means less space for storage and fewer computations in feature matching.

Figure 6. An image used for constructing synthetic underwater images.

Given a clean image S, as shown in Fig. 6, we construct three synthetic images S_1, S_2 and T by applying PSFs with different parameters to S. We then use features on S_1 and S_2 to build the feature codebook and match features between S and T by matching the features on T to the feature codebook. Clearly, we know the ground-truth feature matching: the features at the same locations on T and S should be matched to each other since we do not introduce any spatial transformation in image synthesis. We use two strategies to detect features on S, S_1, S_2 and T.

4.2. First Strategy

In Strategy I, we simply detect SIFT feature points and features on S and then assume the feature points on S_1, S_2 and T are in the same location as the SIFT points detected on S. However, the feature descriptor at each feature point on S_1, S_2 and T is calculated using its own image intensities, but using the optimal scale and orientation parameters derived from the same feature point on S. We apply the three matching algorithms described in Section 3: The classical SIFT matching directly matches features between T and S by using the nearest neighboring technique; the extended SIFT matching matches T and S_i, $i = 1, 2$ independently using classical SIFT matching and then picks the better matching results. The proposed algorithm matches T and the codebook built on S_1 and S_2 by sparse linear combination. We found that, the classical SIFT matching completely fails with a near-zero precision, recall, and F-score. We only compare the performance of the extended SIFT matching and the proposed matching algorithms. We tried different PSF parameters for constructing S_1, S_2 and T and find that the proposed algorithm consistently shows higher precision, recall, and F-score than the extended SIFT matching algorithm. Several sample results are shown in Fig. 7 and 8.

4.3. Second Strategy

In Strategy II, we independently run SIFT feature detection on S, S_1, and S_2. We then take the union of the detected feature points on S, S_1 and S_2. At this set of features points on S_1 and S_2, we extract their SIFT descriptors to build the codebook. On T, we independently run SIFT detection to detect its own SIFT features. We then run the three feature matching algorithms to solve this feature matching problem. Again, we find that classical SIFT matching fails in almost every experiment and the proposed algorithm performs better than the extended SIFT matching algorithm. Samples results are shown in Fig. 7 and 8. Note that the precision, recall, and F-score obtained in Strategy II experiments are much lower than the ones obtained in Strategy I experiments. The reason for this is that many feature points detected in T are not coincident with the feature points detected in S, S_1 and S_2.

We also try to match features between two images with spatial transformations. As shown in Fig. 9, we take two clean images out of water that contain the same object. We then detect SIFT points on both of them. We apply a strategy similar to Strategy I mentioned above where we apply two PSFs to an image shown in Fig. 9(a) to get two synthetic images S_1 ($\tau = 10$, $\omega = 1.0$) and S_2 ($\tau = 10$, $\omega = 0.9$) to construct the feature codebook (feature descriptors are calculated at the same locations as the SIFT points in Fig. 9(a)). We apply a different PSF ($\tau = 10$, $\omega = 0.5$) to the image shown in Fig. 9(b) to construct features T (feature descriptors are calculated at the same locations as the SIFT points in Fig. 9(b)). Figure 9(c) and (d) and show the feature matching results between T and the codebook using the proposed algorithm and the SIFT matching algorithm respectively.

5. Future Work

The proposed approach for feature matching lays the groundwork for performing object detection, localization, and recognition in underwater conditions. But there are still many problems which need to be addressed. First a reliable scheme for detecting feature points that is repeatable in the water is sorely needed. We used a few simple techniques in our experiments but, as shown in the real examples, this approach is far from adequate. Second, better models of the underwater conditions need to be integrated into the approach.

6. Conclusion

In this paper, a method for matching features in underwater environments was proposed. The approach is based on synthesizing possible conditions and looking for linear combinations of these conditions which can explain the ob-
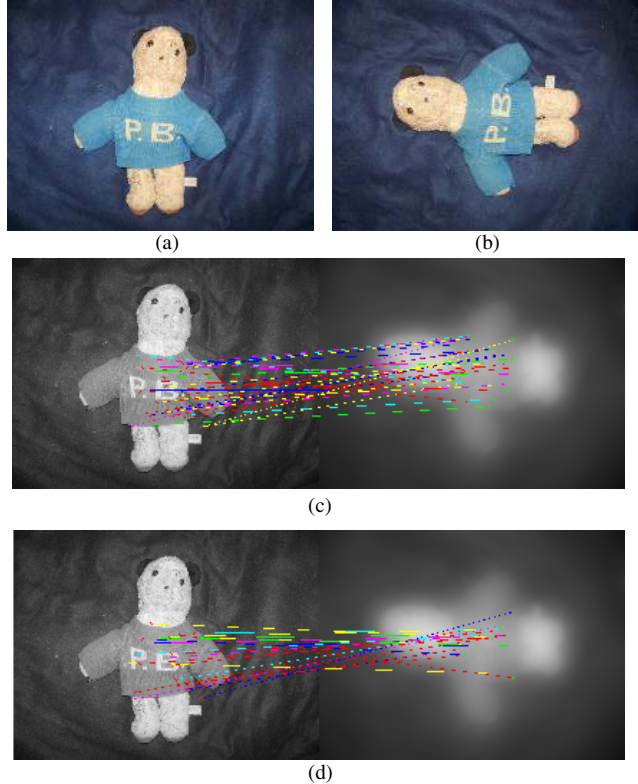

Figure 9. The feature matching results between two images with spatial transformations. (a) and (b) are two clean images. (c) and (d) are the matching results from the proposed algorithm and the extended SIFT matching algorithm, respectively.

served features in an underwater image. Experimental results show that the approach performs better than single nearest neighbor based methods when good feature points are found. However, when detecting features in an underwater image, there is no guarantee that detect features matchable to the out-of-water image can be obtained.

7. Acknowledgments

References

[1] M. Brown, R. Hartley, and D. Nister. Minimal solutions for panoramic stitching. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007. 1

[2] E. Candes, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2), 2006. 4

Codebook

S_1	S_2	T
$\tau = 1 \quad \omega = 0.1$	$\tau = 10 \quad \omega = 1.0$	$\tau = 10 \quad \omega = 0.5$

	SIFT − Ext		Proposed	
	Strat. I	Strat. II	Strat. I	Strat. II
Precision	0.83	0.051	0.785	0.12
Recall	0.249	0.036	0.575	0.187
F−score	0.383	0.042	0.663	0.146

$\tau = 1 \quad \omega = 0.1$	$\tau = 10 \quad \omega = 1.0$	$\tau = 4 \quad \omega = 0.7$

	SIFT − Ext		Proposed	
	Strat. I	Strat. II	Strat. I	Strat. II
Precision	0.983	0.554	0.985	0.612
Recall	0.764	0.35	0.897	0.664
F−score	0.859	0.429	0.939	0.637

$\tau = 1 \quad \omega = 0.1$	$\tau = 10 \quad \omega = 0.5$	$\tau = 10 \quad \omega = 1.0$

	SIFT − Ext		Proposed	
	Strat. I	Strat. II	Strat. I	Strat. II
Precision	0.711	0.0	0.658	0.013
Recall	0.293	0.0	0.484	0.25
F−score	0.415	0.0	0.383	0.026

$\tau = 1 \quad \omega = 0.1$	$\tau = 10 \quad \omega = 0.5$	$\tau = 8 \quad \omega = 0.9$

	SIFT − Ext		Proposed	
	Strat. I	Strat. II	Strat. I	Strat. II
Precision	0.915	0.111	0.924	0.114
Recall	0.542	0.186	0.722	0.377
F−score	0.68	0.139	0.81	0.175

$\tau = 10 \quad \omega = 0.5$	$\tau = 10 \quad \omega = 1.0$	$\tau = 10 \quad \omega = 0.1$

	SIFT − Ext		Proposed	
	Strat. I	Strat. II	Strat. I	Strat. II
Precision	0.607	0.049	0.799	0.087
Recall	0.195	0.028	0.539	0.104
F−score	0.295	0.035	0.643	0.129

$\tau = 10 \quad \omega = 0.5$	$\tau = 10 \quad \omega = 1.0$	$\tau = 4 \quad \omega = 0.6$

	SIFT − Ext		Proposed	
	Strat. I	Strat. II	Strat. I	Strat. II
Precision	0.883	0.139	0.966	0.288
Recall	0.502	0.083	0.913	0.393
F−score	0.64	0.104	0.866	0.332

Figure 7. Results of our experiments. Each row represents one codebook: S_1 and S_2 and test image T, and precision, recall, and F-score table for this image using different methods. SIFT-EXT refers to the approach presented in Section 3.2 and Proposed to the approach in Section 3.3. Strat. I and Strat. II refer to Strategy I and Strategy II as detailed in Section 4.2 and 4.3 respectively.

[3] H. Cheng, Z. Liu, N. Zheng, and J. Yang. A deformable local image descriptor. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. 1

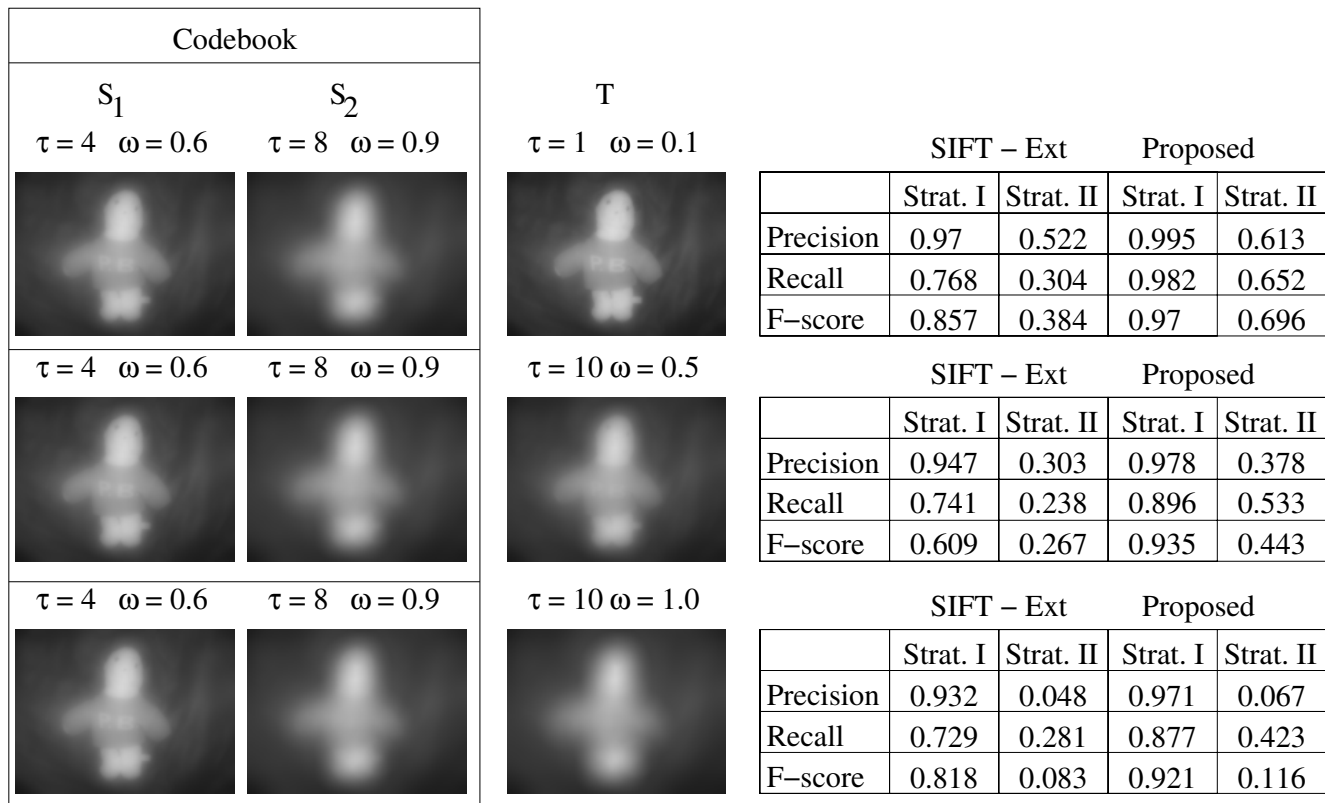[4] L. Dolin, G. Gilbert, I. Levin, and A. Luchinin. *Theory of*

Codebook

S_1	S_2		T
$\tau = 4$ $\omega = 0.6$	$\tau = 8$ $\omega = 0.9$		$\tau = 1$ $\omega = 0.1$

	SIFT − Ext		Proposed	
	Strat. I	Strat. II	Strat. I	Strat. II
Precision	0.97	0.522	0.995	0.613
Recall	0.768	0.304	0.982	0.652
F−score	0.857	0.384	0.97	0.696

S_1	S_2		T
$\tau = 4$ $\omega = 0.6$	$\tau = 8$ $\omega = 0.9$		$\tau = 10$ $\omega = 0.5$

	SIFT − Ext		Proposed	
	Strat. I	Strat. II	Strat. I	Strat. II
Precision	0.947	0.303	0.978	0.378
Recall	0.741	0.238	0.896	0.533
F−score	0.609	0.267	0.935	0.443

S_1	S_2		T
$\tau = 4$ $\omega = 0.6$	$\tau = 8$ $\omega = 0.9$		$\tau = 10$ $\omega = 1.0$

	SIFT − Ext		Proposed	
	Strat. I	Strat. II	Strat. I	Strat. II
Precision	0.932	0.048	0.971	0.067
Recall	0.729	0.281	0.877	0.423
F−score	0.818	0.083	0.921	0.116

Figure 8. More results of our experiments.

imaging Through Wavy Sea Surface. Russian Academy of Sciences, Institute of Applied Physics, Nizhniy Novgorod, 2006. 2, 3

[5] D. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4), 2006. 4

[6] R. Hess and A. Fern. Improved video registration using non-distinctive local image features. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007. 1

[7] W. Hou. A simple underwater imaging model. *Optics Express*, 34(17), 2009. 2

[8] W. Hou, D. Gray, A. Weidemann, and R. Arnone. Comparison and validation of point spread models for imaging in natural waters. *Optics Express*, 16(13), 2008. 2, 3

[9] J. Jaffe. Monte carlo modeling of underwate-image formation: Validity of the linear and small-angle approximations. *Applied Optics*, 34(24):5413–5421, 1995. 2

[10] H. Jin. A three-point minimal solution for panoramic stitching with lens distortion. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. 1

[11] Y. Ke and R. Sukthankar. Pca-sidt: A more distinctive representation for local image descriptors. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 506–513, 2005. 1

[12] C. Lampert, M. Blaschko, and T. Hofmann. Efficient subwindow search: A branch and bound framework for object

localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 2129–2142, December 2009. 1

[13] H. Ling and D. Jacobs. Deformation invariant feature matching. In *International Conference on Computer Vision*, pages 1466–1473, 2005. 1

[14] D. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision*, pages 1150–1157, 1999. 1

[15] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal on Computer Vision*, 60(2):91–110, 2004. 1, 3

[16] G. Medioni. Retinal image registration from 2d to 3d. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. 1

[17] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005. 1

[18] C. Mobley. *Light and Water: Radiative Transfer in Natural Waters*. Academic Press, 1994. 2

[19] E. N. Mortensen, D. Hongli, and L. Shapiro. A sift descriptor with global context. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 184–190, 2005. 1

[20] A. Torralba, K. Murphy, and W. Freeman. Sharing visual features for multiclass and multiview object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 854–869, May 2007. 1