The Misty Three Point Algorithm for Relative Pose

Tobias Palmér, Kalle Åström
Center for Mathematical Sciences
Lund University
{tobiasp,kalle}@maths.lth.se

Jan-Michael Frahm
Department of Computer Science
The University of North Carolina at Chapel Hill
jmf@cs.unc.edu

Abstract

There is a significant interest in scene reconstruction from underwater images given its utility for oceanic research and for recreational image manipulation. In this paper we propose a novel algorithm for two view camera motion estimation for underwater imagery. Our method leverages the constraints provided by the attenuation properties of water and its effects on the appearance of the color to determine the depth difference of a point with respect to the two observing views of the underwater cameras. Additionally, we propose an algorithm, leveraging the depth differences of three such observed points, to estimate the relative pose of the cameras. Given the unknown underwater attenuation coefficients, our method estimates the relative motion up to scale. The results are represented as a generalized camera. We evaluate our method on both real data and simulated data.

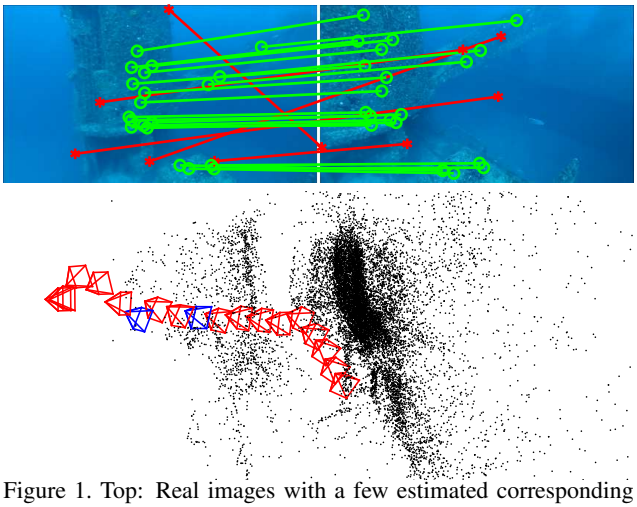

Figure 1. Top: Real images with a few estimated corresponding points (green) and outliers (red) found using the Misty Three Point algorithm (MTP) proposed in this paper. Bottom: structure and motion estimated using MTP for a sequence (in which the top row images are represented by blue cameras).

1. Introduction

The recovery of 3D scene geometry from images has always been one of the core goals of computer vision and has progressed significantly over the last decade [19, 1, 5, 9]. One essential building block of all these system is the ability to successfully estimate the two-view geometry between two overlapping cameras under the assumption of a central perspective camera. This camera model though is only valid for cameras taking photos through air. Hence, this mode of reconstruction is not valid for cameras submerged in water. However, 71% of the earth's surface is covered by oceans and with both the scientific interest in underwater imagery and the now ubiquitous availability of underwater cameras to users, for example through GoPro cameras, performing structure from motion in underwater environments moves into the focus of research [18, 12]. In this paper we target the successful estimation of two-view geometry for underwater cameras, which due to their imaging conditions do not comply with the traditionally used pinhole camera model [18, 22]. In this paper we propose a novel method to enable SFM under such circumstances. In particular we propose a novel minimal solver to enable two-view geometry estimation and a method for relative point depth estimation.

Our proposed method leverages the observation that when looking at photos for example of an under water shipwreck, one can get a sense of depth. The reason for this is the easily observable depth dependent attenuation of light under water, which is more significant than in air and weakens as well as distorts an object's color [15, 16, 21].We leverage this observation along with the 2D correspondences of the projections of the same 3D scene point into two different images to deduce a novel constraint on the relative depth change of the point with respect to the two capturing cameras (see Section 2.3). Given the relative depth changes we then propose a novel three point algorithm that leverages these changes in order to infer the relative camera motion up to scale (see Section 3). In combination this enables us to obtain the two view geometry under a gen-

eralized camera model between two images of intrinsically calibrated underwater cameras. Next we discuss the related work in the area of underwater image based pose estimation.

1.1. Related work

Agrawal et al. [2] present theory and methods for multi-layer flat refractive scenes, using an arbitrary number of interfaces. A method for calibrating such a system is presented, and multi-layer systems are shown to be well approximated by single/two layer systems. A similar method for calibration is proposed by Treibiz et al. [22]

A few special cases encountered in flat refractive geometry are solved using polynomial solvers and geometric algebra [6]. They develop near-minimal solvers for the general calibrated and unknown focal length absolute pose cases, i.e. for cases where the scene coordinates are assumed to be known. In this paper we use similar polynomial solving techniques but with unknown scene points.

In the field of underwater structure from motion, Sedlazeck et al. [17] have created a system for simulating deep sea underwater images using physical models for light attenuation, scattering and refraction. Furthermore, Sedlazeck et al. have shown that approximating underwater structure from motion by pinhole cameras produces a systematic error [18]. They propose a method for underwater structure from motion using a virtual camera model [11, 12]. Jordt-Sedlazeck [11] provides a broader overview of the theory and the field of underwater SFM. Their proposed methods uses at least five points for pose estimation (assuming the camera is intrinsically calibrated) while our method only uses three. This is important in a number of scenarios – the most obvious being that our algorithm would on average require fewer RANSAC-iterations to estimate camera motion.

Schechner et al. [15] propose to recover the scene object radiance, and as a by-product the relative distances in the scene are estimated and yield range maps for the scene. These relative distances are used as a ratio of improvement of the visibility range achieved by the recovery method. Furthermore, Schechner et al. [16] reconstruct dense 3-D images of the scene using the depth map and the recovered image. Their approach treats depth maps as a by-product of estimating the scene radiance, whereas our proposed method provides a fusion of their two separate methods. Lastly, Swirski and Schechner [21] propose a method to remove another underwater disturbance – *flicker*.

Queiroz et al. [14] uses the same physical imaging model as in this paper, and leverages color information to improve dense stereo maps. However, they assume a manual pre-calibration of all underwater imagery parameters as well as both geometric and radiometric pre-calibration. The physical imaging model is only used for adding an automated penalty function for the depth map estimation. The algorithm is later automatized by Nascimento et al. [13], but the physical imaging model is still not used for camera pose estimation. Our method, however, automatically estimates the underwater imagery parameters and uses the physical imaging model for pose estimation.

A problem that is related to restoration of underwater images is haze removal. Some methods propose to model attenuation in haze using the same physical model as in this paper. This creates an underconstrained problem in single images. Fattal [4] proposes to assume that surface shading is locally uncorrelated to the transmission function in order to add constraints. A different constraint, the *dark channel prior*, is proposed by He et al. [8]. Bahat and Irani propose to use the recurring patch property as a prior [3]. All of these methods estimate depth images, but they are only treated as by-products in the process of the image restoration and not used as constraints to estimate (relative) camera pose.

To our knowledge, optimal two-view structure-and-motion using three points and known depth differences has not been solved to date. Neither has the problem of two-view generalized camera structure-and-motion using three points and their colors.

1.2. Innovations

In particular, we propose the following novel methods:

- A minimal solver, the Three Point Delta algorithm (TPD), for estimating the relative motion of a generalized camera given three pairs of point correspondences and differences in depth for each such correspondence.
- The Three Colors Depth Difference algorithm (TCDD), for using a physical model of light propagation under water to estimate relative depth differences.
- The Misty Three Point algorithm (MTP), for estimating the relative motion of a generalized camera given three pairs of point-and-color correspondences.

Moreover, we show that the methods perform very well numerically and are stable to large amounts of outliers when implemented within a RANSAC-framework, which is enabled by the fact that it only requires 3 points for estimating the relative motion. Furthermore, the requirement of only three points is also useful when there are only three or four points available. In these cases, our proposed methods can estimate the relative motion while previous methods can not.

1.3. Additional applications

Although the presented minimal solver (TPD) is applied on underwater images in this paper, we note a few more potential areas of application. For example, it can be used

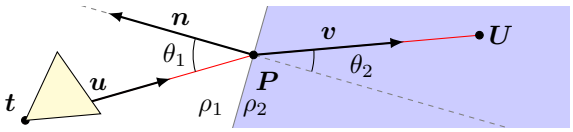

Figure 2. Snell's law. A ray originating from the camera center $\boldsymbol{t}$ with direction $\boldsymbol{u}$ changes direction at $\boldsymbol{P}$ according to $\rho_1 \sin\theta_1 = \rho_2 \sin\theta_2$. This causes the usually linear equations for projections, for example, to become nonlinear and much harder to solve.

to estimate relative motion using both images and sound. Or to estimate the relative motion of microphone arrays. Furthermore, the system could theoretically be used in the case of above-ground structure from motion in the presence of fog (in which the attenuation of light can be modeled similarly to water). It is also possible to use together with pseudo depth estimation parts of previous methods for haze-removal, e.g. [3]. In fact, the relative pose estimation algorithms can be applied for any setting (once again assuming pre-calibration) given three point correspondences and distance differences for each pair.

2. Estimating relative depth in underwater imagery

The physical conditions for imaging under water are significantly different than in air leading to a distinct set of challenges for computer vision methods.

One of the main differences is the nonlinear geometry [2, 12, 11] - as cameras usually need to be enclosed in protective housing, which changes the direction of the light rays. In air, rays from the source travel in straight lines to the camera lens. In water, rays are refracted at the surface of the (usually flat) port of the underwater housing. Given that the orientation and position of the refractive surface relative to the camera is known, Snell's law (see Fig 2) can be used to determine the outgoing rays from a point on the camera sensor. It states that the angle θ_1, relative to the normal of the interface, of an incident ray is related to the angle of the refracted ray θ_2 by the equation

$$\rho_1 \sin\theta_1 = \rho_2 \sin\theta_2, \tag{1}$$

where ρ_1 and ρ_2 are the refractive indices of the two media. The fact that this equation is nonlinear is one of the main causes of the challenge in the field of underwater structure from motion. For example, finding the ray in the scene corresponding to an observed pixel is not much more difficult than for regular cameras. Note, though, that it requires that the intrinsics of the camera and the relation to and geometry of the underwater housing is known. However, the reversed problem of finding the projection of a point in space on the image plane is significantly more challenging than for regular cameras.

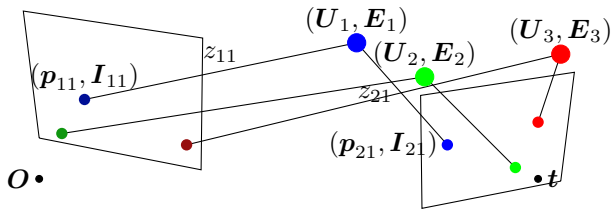

Figure 3. The relative pose problem solved in this paper. Two cameras positioned at the origin and $\boldsymbol{t}$, respectively, are observing three points of unknown position and unknown color. Note that we are not only using the position and direction $(\boldsymbol{p}_{ik}, \boldsymbol{u}_{ik})$ from each observed point, but also the observed color. The relative depths $\Delta z_k = z_2 - z_1$ are also crucial parts of this method. In the left cameras, with larger distance to the objects, the colors look very similar. In the right camera, however, the observed colors are more similar to the actual color of the object. These differences is what enables the estimation of depth differences.

Another significant difference is the appearance - colors look different under water than when imaged through air [10, 15, 16]. A visualization of this is presented in Fig. 4. There are two sources of light that can be observed: light from the scene object and light as part of the ambient light. The observed ambient light part will be termed the *veiling light*. In this paper, the observation of the veiling light is caused by ambient light being scattered towards the cameras line of sight by the particles in the water. The light signal from the scene object can by itself be seen as composed of two components - direct transmission and forward scattering. Whereby the direct transmission is the signal after losing energy due to particles absorbing and scattering (in all directions) the light from the scene object, causing an exponential decrease in energy. The forward scattering is caused by particles scattering the light in small angles relative to the line of sight, causing a blur as well a decrease in energy. A crucial part regarding attenuation is that different wavelengths are attenuated at different rates - red is absorbed 10 times faster than green, which in pure water is absorbed approximately twice as fast as blue. This causes the natural ambient light to be blue/green since the red/yellow components of the natural sunlight are absorbed too quickly to be perceived.

In the remainder of this section, we will present our novel method for taking advantage of this difference in observed color, by assuming that a scene object will have the same unknown radiance when viewed in different images. We then introduce how the observed difference in color between observations of a scene point can be used to estimate differences in distance to the point.

2.1. The physical model

The Jaffe-McGlamery equation is a commonly used equation that models the effect of absorption and scatter-

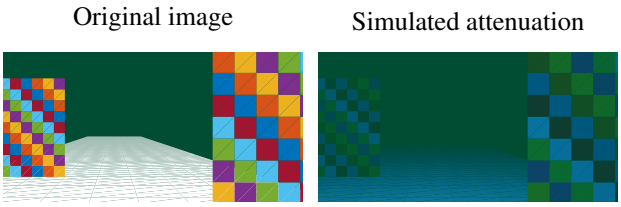

Figure 4. A visualization of the imagery effects of attenuation. The left image shows a generated scene, and the right image shows the same scene after simulating attenuation of colors by Eq. (2).

ing [10]. In this paper, a simplified version of the equation (Eq. (2)) is used, which does not take the forward scattering into account. The forward scattering can be neglected as the dominant cause for image contrast degradation is the veiling light [15, 16], and furthermore a large part of the forward scattering can be seen as attenuation, which is already captured in the simplified model (Eq. (2)).

First, we introduce our notation. The point in space with index k is represented by $(\boldsymbol{U}_k, \boldsymbol{E}_k)$, where the three-vector $\boldsymbol{U}_k$ is the position and the three-vector $\boldsymbol{E}_k = (E_k^r, E_k^g, E_k^b)$ is the radiance of the object in the color channels $\lambda \in \{r, g, b\}$. The observation of the point with index k in camera i is represented in the coordinate system of camera i by $(\boldsymbol{u}_{ik}, \boldsymbol{I_{ik}}, \boldsymbol{p}_{ik})$, where $\boldsymbol{u}_{ik}$ is the direction of the ray in water, $\boldsymbol{p}_{ik}$ is a point on the ray and $\boldsymbol{I_{ik}} = (I_{ik}^r, I_{ik}^g, I_{ik}^b)$ is the observed color. Note that in the case of the pinhole camera model, all $\boldsymbol{p}_{ik} = \boldsymbol{0}$, and in the case of the generalized camera model to account for refractions, the $\boldsymbol{p}_{ik}$ are points on the refractive plane. The depths z_{ik} denote the Euclidean distance from $\boldsymbol{p}_{ik}$ to $\boldsymbol{U}_k$, i.e. $z_{ik} = \|\boldsymbol{p}_{ik} - (R\boldsymbol{U}_k - R\boldsymbol{t})\|_2$, where R is a rotation matrix and $\boldsymbol{t}$ a translation vector that together transform scene points to the local coordinate system of the camera.

The equation used for modeling the physical effects towards the observed colors is the simplified version of the Jaffe-McGlamery equation

$$I_{ik}^\lambda = \alpha_\lambda \left(E_k^\lambda e^{-\eta_\lambda z_{ik}} + B_\infty^\lambda (1 - e^{-\eta_\lambda z_{ik}}) \right) + \beta_\lambda, \quad (2)$$

where I_{ik}^λ is the pixel value in color channel λ for camera i and point k, α_λ and β_λ are color correction coefficients [11], E_k^λ is the "true" but unknown color of point k, η_λ is the attenuation coefficient of the water, z_{ik} is the distance from the outer refraction plane of camera i to point k and B_∞^λ is the "veiling light". This is a convex combination of the true color and the veiling light, with added color correction. Note that we here assume that the two cameras have the same color correction coefficients. The equation can be reformulated to:

$$I_{ik}^\lambda = \alpha_\lambda \left(E_k^\lambda - B_\infty^\lambda \right) e^{-\eta_\lambda z_{ik}} + \gamma_\lambda, \quad (3)$$

where $\gamma_\lambda = \alpha_\lambda B_\infty^\lambda + \beta_\lambda$ is the observation of the veiling light. Given observations $\{I_{1k}^\lambda\}$ and $\{I_{2k}^\lambda\}$, $\lambda \in \{g, b\}$, of the color of point k in cameras 1 and 2 at depth z_{1k} and z_{2k}, the equation can be reduced to

$$\frac{I_{1k}^\lambda - \gamma_\lambda}{I_{2k}^\lambda - \gamma_\lambda} = \frac{\alpha_\lambda \left(E_k^\lambda - B_\infty^\lambda \right) e^{-\eta_\lambda z_{1k}}}{\alpha_\lambda \left(E_k^\lambda - B_\infty^\lambda \right) e^{-\eta_\lambda z_{2k}}} = e^{\eta_\lambda \Delta z_k}. \quad (4)$$

2.2. Estimating the constant parameters

The red color channel is not used due to the fact that in practical applications the red colors are by practical means completely absent, as discussed in Section 2. Accordingly, there are 4 unknown constant parameters to estimate, and each pairwise correspondence introduces 1 unknown variable (Δz_k) and provides two constraints by using only the green and blue color channels in Eq. (4). This means that to solve for all $4+n$ unknown constants and variables, at least $n = 4$ pairwise correspondences are needed. However, a scale ambiguity exists, thus η_λ and z_{ik} can be redefined as $\hat{\eta}_\lambda = \eta_\lambda / \eta_g$ and $\hat{z}_{ik} = \eta_g z_{ik}$. Thus for each point correspondence we have the constraints

$$\begin{aligned} I_{1k}^g - \gamma_g &= (I_{2k}^g - \gamma_g) e^{\Delta \hat{z}_k}, \\ I_{1k}^b - \gamma_b &= (I_{2k}^b - \gamma_b) e^{\hat{\eta}_b \Delta \hat{z}_k}. \end{aligned} \quad (5)$$

Furthermore, solving for γ_λ and reformulating Eq. (5) leads to

$$\gamma_\lambda = I_{2k}^\lambda + \frac{1}{1 - e^{\hat{\eta}_b \Delta \hat{z}_k}} \left(I_{1k}^\lambda - I_{2k}^\lambda \right), \quad (6)$$

which is monotonous in $\Delta \hat{z}_k$. This monotony can be used to show that there does not always exist a real solution $(\Delta \hat{z}_k, \gamma_g, \gamma_b, \hat{\eta}_b)$.

2.3. The Three Colors Depth Difference algorithm

After introducing the obtained constraint for each point correspondence we now introduce our proposed depth estimation. We define the underwater error function for the color channel λ as (rewritten from Eq. (4))

$$r_\lambda(\gamma_\lambda, \eta_\lambda, \Delta z_k) = I_{1k}^\lambda - \gamma_\lambda - (I_{2k}^\lambda - \gamma_\lambda) e^{\eta_\lambda \Delta z_k}, \quad (7)$$

and the combined error function for the green and blue channel as $r(\boldsymbol{\gamma}, \boldsymbol{\eta}, \Delta z_k) = r_g^2 + r_b^2$. The Jacobian for $r(\boldsymbol{\gamma}, \boldsymbol{\eta}, \Delta z_k)$ is computed and used in a Gauss-Newton algorithm to find the parameters for which the error is minimized.

Note that if $I_{1k}^\lambda - \gamma_\lambda$ has a different sign than $I_{2k}^\lambda - \gamma_\lambda$ there is no solution to $r_\lambda = 0$ (since $e^x > 0$), and the minimum is found at $\Delta z_k = 0$. This would correspond to a point being observed beyond infinity. Thus it can be concluded that

$$\min_{\Delta z} r_\lambda^2(\gamma_\lambda, \eta_\lambda, \Delta z) \geq (I_{1k}^\lambda - \gamma_\lambda)^2 \quad (8)$$

This means that a point in one camera with, for example, green intensity larger than γ_g will still have green intensity

larger than γ_g when observed in another camera. This lower bound will later be used for fast outlier rejection when implementing the three-point relative pose algorithm within RANSAC (see Section 4.1).

3. The Three Point Delta algorithm

In this section we propose a novel method to compute the relative motion between two intrinsically calibrated generalized cameras given image point observations of three scene points in both cameras, as well as the *differences* in distance from the cameras to the scene points. We formulate the problem as a system of polynomial equations in the unknown absolute distances to the first camera. This system is solved by the action matrix method (see e.g. [20]), which entails reformulating the system as an eigenvalue equation. Code for this is provided in the supplementary material. Once the absolute distances to the first camera have been computed, the scene points can be reconstructed in the local coordinate system. Then the pose of the second camera is computed by the three point resection method [7]. The details of our method are outlined in the following.

3.1. The algorithm

Assume that three pairs of point correspondences, $\{\boldsymbol{x}_{1k}\}$ and $\{\boldsymbol{x}_{2k}\}$, in two intrinsically calibrated cameras are given. The relation between the local coordinate system of the first camera and the global coordinate system can w.l.o.g. be fixed to $(I, \mathbf{0})$. Then the relative pose of the second camera is described by the sought rigid transformation $(R, \boldsymbol{t})$. This means that for each observed scene point $\boldsymbol{U}_k$, we have the ray parametrizations

$$\begin{aligned} \boldsymbol{p}_{1k} + z_{1k}\boldsymbol{u}_{1k}, &\qquad z_{1k} \in \mathbb{R}, \\ R^\top(\boldsymbol{p}_{2k} + z_{2k}\boldsymbol{u}_{2k}) + \boldsymbol{t}, &\qquad z_{2k} \in \mathbb{R}, \end{aligned} \tag{9}$$

where $\boldsymbol{p}_{1k}$, $\boldsymbol{p}_{2k}$, $\boldsymbol{u}_{1k}$ and $\boldsymbol{u}_{2k}$ are known since the cameras are intrinsically calibrated, and z_{1k}, z_{2k}, R and $\boldsymbol{t}$ are the sought parameters. Furthermore, we assume that the difference in distance to each scene point $\boldsymbol{U}_k$, i.e. $\Delta z_k = \|\boldsymbol{U}_k - \boldsymbol{p}_{1k}\| - \|\boldsymbol{U}_k - (R^\top \boldsymbol{p}_{2k} + t)\|$, is known.

Note that in the particular case where the generalized camera is a pinhole camera, an unknown scale of depth differences simply gives a scale ambiguity in the solution. In the general case, an unknown scale of depth differences make the solution non-existing or invalid. Furthermore, note that in the case where the generalized cameras are cameras enclosed in underwater housings, the $\boldsymbol{p}_{1k}$'s and $\boldsymbol{p}_{2k}$'s are points on the outer surfaces of the underwater housing ports, and the $\boldsymbol{u}_{1k}$'s and $\boldsymbol{u}_{2k}$'s are the directions into the water.

Since we assume that x_{1k} and x_{2k} correspond to the same scene point, there exists z_{1k} and z_{2k} such that $\boldsymbol{p}_{1k} + z_{1k}\boldsymbol{u}_{1k} = R^\top(\boldsymbol{p}_{2k} + z_{2k}\boldsymbol{u}_{2k}) + \boldsymbol{t}$ (using the parametrization from Eq. (9)). The unknown depths z_{2k} can be reduced by substituting $z_{2k} = z_{1k} + \Delta z_k$, which gives the sets of equations

$$\boldsymbol{p}_{1k} + z_{1k}\boldsymbol{u}_{1k} = R^\top(\boldsymbol{p}_{2k} + (z_{1k} + \Delta z_k)\boldsymbol{u}_{2k}) + \boldsymbol{t}, \quad k = 1, 2, 3, \tag{10}$$

where depths z_{1k} and the relative pose $(R, \boldsymbol{t})$ of the second camera are the unknown variables that are sought. We propose to parametrize R using quaternions $q = (s, \boldsymbol{v})$, where s is scalar and $\boldsymbol{v}$ is a three-vector,

$$R = 2(\boldsymbol{v}\boldsymbol{v}^\top - s[\boldsymbol{v}]_\times) + (s^2 - \boldsymbol{v}^\top\boldsymbol{v})I, \tag{11}$$

and adding the constraint $s^2 + \boldsymbol{v}^\top\boldsymbol{v} = 1$ to ensure that the determinant is one, gives a total of 10 equations in 10 unknowns.

The equations are solved by noting that the points $\boldsymbol{U}_k = \boldsymbol{p}_{1k} + z_{1k}\boldsymbol{u}_{1k}$ and $\boldsymbol{U}'_k = \boldsymbol{p}_{2k} + (z_{1k} + \Delta z_k)\boldsymbol{u}_{2k}$ are related by a rigid transform. Thus the Gramians for the two sets of points, $\{\boldsymbol{U}_k\}$ and $\{\boldsymbol{U}'_k\}$, are equal [23]. The Gramian for U is defined as $V^\top V$, where

$$V = \begin{bmatrix} \boldsymbol{U}_2 - \boldsymbol{U}_1, & \boldsymbol{U}_3 - \boldsymbol{U}_1 \end{bmatrix}. \tag{12}$$

Inserting the expressions for the $\boldsymbol{U}_k$'s and $\boldsymbol{U}'_k$'s gives the Gramians

$$\begin{aligned} V &= \begin{bmatrix} z_{12}\boldsymbol{u}_{12}^\top - z_{11}\boldsymbol{u}_{11}^\top + \boldsymbol{p}_{12}^\top - \boldsymbol{p}_{11}^\top \\ z_{13}\boldsymbol{u}_{13}^\top - z_{11}\boldsymbol{u}_{11}^\top + \boldsymbol{p}_{13}^\top - \boldsymbol{p}_{11}^\top \end{bmatrix}^\top, \\ V' &= \begin{bmatrix} (z_{12}+\Delta z_2)\boldsymbol{u}_{22}^\top - (z_{11}+\Delta z_1)\boldsymbol{u}_{21}^\top + \boldsymbol{p}_{22}^\top - \boldsymbol{p}_{21}^\top \\ (z_{13}+\Delta z_3)\boldsymbol{u}_{23}^\top - (z_{11}+\Delta z_1)\boldsymbol{u}_{21}^\top + \boldsymbol{p}_{23}^\top - \boldsymbol{p}_{21}^\top \end{bmatrix}^\top. \end{aligned} \tag{13}$$

Thus the constraint that the Gramians are equal amounts to

$$V^\top V - (V')^\top(V') = 0, \tag{14}$$

and provides three equations that are quadratic in the three unknowns z_{1k}.

It turns out that the coefficients for all z_{1k}^2 terms are zero since the $\boldsymbol{u}_{1k}$'s and $\boldsymbol{u}_{2k}$'s are normalized. Thus the equations are of the form

$$\begin{cases} A_{11}xy & +A_{14}x + A_{15}y & +A_{17} = 0, \\ A_{21}xy + A_{22}xz + A_{23}yz + A_{24}x + A_{25}y + A_{26}z + A_{27} = 0, \\ A_{32}xz & +A_{34}x & +A_{36}z + A_{37} = 0, \end{cases} \tag{15}$$

where x, y and z correspond to z_1, z_2 and z_3. The system of equation (15) can be represented as a matrix-vector multiplication $A\boldsymbol{v} = \mathbf{0}$, where

$$A = \begin{bmatrix} A_{11} & 0 & 0 & A_{14} & A_{15} & 0 & A_{17} \\ A_{21} & A_{22} & A_{23} & A_{24} & A_{25} & A_{26} & A_{27} \\ 0 & A_{32} & 0 & A_{34} & 0 & A_{36} & A_{37} \end{bmatrix}, \tag{16}$$

and $\boldsymbol{v}$ is the vector of monomials

$$\boldsymbol{v} = (xy, xz, yz, x, y, z, 1). \tag{17}$$

By performing row-operations on the system, it can be simplified to the form $\hat{A}\boldsymbol{v} = \mathbf{0}$, where

$$\hat{A} = \begin{bmatrix} 1 & 0 & 0 & \hat{A}_{14} & \hat{A}_{15} & 0 & \hat{A}_{17} \\ 0 & 1 & 0 & \hat{A}_{24} & 0 & \hat{A}_{26} & \hat{A}_{27} \\ 0 & 0 & 1 & 0 & \hat{A}_{35} & \hat{A}_{36} & \hat{A}_{37} \end{bmatrix}. \tag{18}$$

To solve the system, x times the third equation, y times the second equation and z times the third equation are added, i.e. the equations

$$\begin{cases} xyz + C_{14}xz & +C_{15}yz + C_{17}z & = 0, \\ xyz + C_{24}xy & +C_{26}yz + C_{27}y & = 0, \\ xyz + C_{35}xy & +C_{36}xz + C_{37}x & = 0, \end{cases} \tag{19}$$

are added. Since Eq. (18) gives reductions from xy, xz and yz to x, y and z, xyz can also be reduced to x, y and z. Thus the system can be formulated as

$$\begin{cases} M_{11}y + M_{12}z + M_{13} = xy, \\ M_{21}y + M_{22}z + M_{23} = xz, \\ M_{31}y + M_{32}z + M_{33} = x, \end{cases} \tag{20}$$

i.e. as the eigenvalue equation

$$M \begin{bmatrix} y & z & 1 \end{bmatrix}^\top = x \begin{bmatrix} y & z & 1 \end{bmatrix}^\top. \tag{21}$$

Thus the depth z_{11} is an eigenvalue of M, and z_{12} and z_{13} are the two first elements of the corresponding eigenvector after normalizing with the third element. Since M is a 3 by 3 matrix, three solutions are found and need to be evaluated by Eq. (15).

Assuming that one solution (z_{11}, z_{12}, z_{13}) to Eq. (15) was found, the z_{2k}'s can be computed by $z_{2k} = z_{1k} + \Delta z_k$. Now that all depths are known, the $\boldsymbol{U}_k$'s can be computed from the z_{1k}'s as in $\boldsymbol{U}_k = \boldsymbol{p}_{1k} + z_{1k}\boldsymbol{u}_{1k}$. Eq. (10) gives that $\boldsymbol{U}_k = R^\top(\boldsymbol{p}_{2k} + z_{2k}\boldsymbol{u}_{2k}) + \boldsymbol{t}$ where R and $\boldsymbol{t}$ are the only remaining unknowns, which means that the remaining problem is now to find a projective transformation from $\boldsymbol{U}_k$ to $\boldsymbol{p}_{2k} + z_{2k}\boldsymbol{u}_{2k}$. This problem is solved by the three point resection method [7], providing up to four solutions for $(R, \boldsymbol{t})$. All solutions $(R, \boldsymbol{t})$ that give negative depths when projecting the $\boldsymbol{U}_k$'s are rejected. Then the distance from each point $\boldsymbol{p}_{2k}$ to the corresponding transformed scene point $R\boldsymbol{U}_k - R\boldsymbol{t}$ is computed as

$$z'_{2k} = \|(R\boldsymbol{U}_k - R\boldsymbol{t}) - \boldsymbol{p}_{2k}\|_2. \tag{22}$$

If the relative pose $(R, \boldsymbol{t})$ is valid, then $z_{2k} = z'_{2k}$ must hold. Thus, for each solution (z_{11}, z_{12}, z_{13}) to Eq. (15) there are at most four solutions $(R, \boldsymbol{t})$, which makes a total of up to twelve solutions to be evaluated as described.

In conclusion, we have shown how given only three pairs of corresponding points and their difference in distance to the camera, first the absolute distances can be found and subsequently the object and the relative motion. Furthermore, by combining this method with the method for estimating relative depths given the colors of three pairs of corresponding points, we have found a method for estimating relative motion of a generalized camera given three corresponding points with their associated colors in the images.

4. Robust estimation with the Misty Three Point algorithm

In this section we show how the Misty Three Point algorithm (MTP) can be embedded in a RANSAC-framework using a sequencing of the algorithm that enables fast rejection of an estimate as well as fast rejection of outliers. Given a set of inliers, we also show how the solution can be optimized for all parameters while taking both reprojection errors and the physical model for underwater imaging into account.

4.1. RANSAC

Since the Three Point Delta algorithm (TPD) introduced in Section 3 uses relative depths as input, the Three Colors Depth Difference algorithm (TCDD) defined in Section 2.3 must be the first step of relative pose estimations. In general, the underwater imaging parameters that TCDD provides are not necessarily feasible for all points, giving an opportunity to ignore those points in the following steps. In addition, there is not always a solution to the depth difference problem, in those cases the estimate based on those points can be instantly rejected. Assuming that the algorithm found feasible parameters, TPD is then used to estimate the relative pose. At this point, the scene points are reconstructed by triangulation, and the reprojection errors are computed and used to evaluate how well the estimated pose fits the dataset. Furthermore, the depth differences are also estimated for all feasible points, adding one more measure of how well the estimate fits the dataset.

4.2. Bundle Adjustment

The results from the RANSAC method are optimized in a bundle adjustment algorithm that seeks to minimize the errors both for the reprojections and the underwater imagery equations using all variables. In particular, it optimizes for the relative motion of the generalized cameras and the 3-D points. We define the residual vector, whose norm is the target for minimization, by combining the reprojection errors

with Eq. (3):

$$\boldsymbol{r}(\boldsymbol{x},\boldsymbol{\theta}) = \begin{bmatrix} \dots \\ I^g_{1k} - \hat{E}^g_k e^{-\eta_g z_k} - \gamma_g \\ I^b_{1k} - \hat{E}^b_k e^{-\eta_b z_k} - \gamma_b \\ I^g_{2k} - \hat{E}^g_k e^{-\eta_g z'_k} - \gamma_g \\ I^b_{2k} - \hat{E}^b_k e^{-\eta_b z'_k} - \gamma_b \\ \boldsymbol{U}_k - z_k \boldsymbol{u}_k \\ R\boldsymbol{U}_k - R\boldsymbol{t} - z'_k \boldsymbol{u}'_k \\ \dots \\ s^2 + \boldsymbol{v}^\top \boldsymbol{v} - 1 \end{bmatrix} \tag{23}$$

where radiance $\hat{E}^\lambda_k = \alpha(E^\lambda_k - B^\lambda_\infty)$ and R is parametrized as in Eq. (11). The partial derivatives of $\boldsymbol{r}$ are computed analytically and are used for minimizing $\|\boldsymbol{r}(\boldsymbol{x},\boldsymbol{\theta})\|$ using Gauss-Newton's method with respect to $\boldsymbol{x}$.

5. Experiments

We evaluate our method on simulated and real data. First, we show that the Three Point Delta algorithm (TPD; see Section 3) is numerically stable with respect to estimating the depth. Secondly, we show that the Three Colors Depth Difference algorithm (TCDD; see Section 2.3) is numerically stable. Thirdly, we show that the Misty Three Point Algorithm (MTP; see Section 4), which is the sequential combination of TCDD and TPD, is also numerically stable. Lastly, we show that the RANSAC-based algorithm introduced in Section 4.1 handles outliers well.

The simulated experiments are then complemented with real data experiments. Comparisons are made with a manually obtained baseline. In addition, real experiments are presented where we do not have ground truth to compare with, hence, we compare them only qualitatively.

5.1. Synthetic data

In order to test the system, we generate an underwater scene and observe it by two cameras. That is, the pose of the first camera is fixed to $(I, \mathbf{0})$ random values are generated for: the relative pose $(R, \boldsymbol{t})$ of the second camera, three 3D-points with RGB-colors and underwater imaging parameters $((\gamma_g, \gamma_b, \hat{\eta}_b)$; described in Section 2). The 3D-points were projected into the cameras and the known distances from the 3D-points to the cameras were used to find the attenuated colors according to Eq. (2).

The generated underwater scene was used to test the accuracy of the proposed methods (MTP, TPD and TCDD) as follows. MTP was used to estimate the relative pose, using the projected points and the observed colors. The relative translational error and the relative angular error for the estimate were computed and are presented in Fig 5. Similarly, TPD was used to estimate the relative pose using the projected points and the given difference in depth, the results of which are also presented in Fig 5. These results show

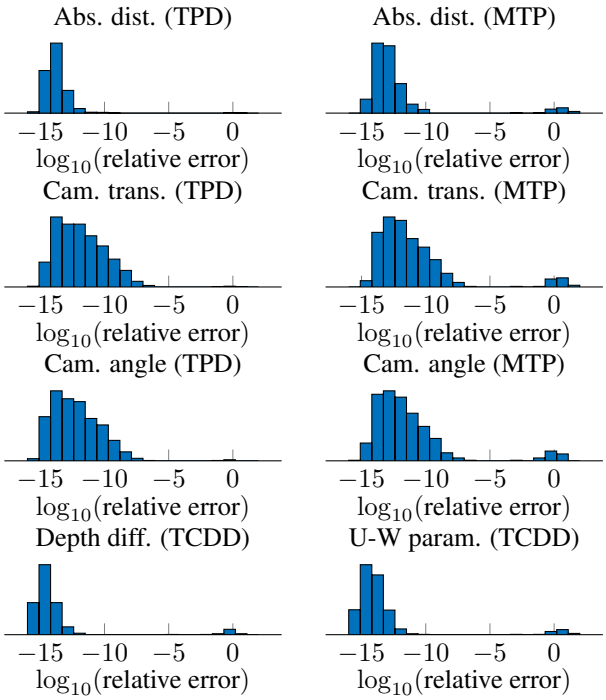

Figure 5. Distribution of solver error relative to ground truth, computed over 10000 random problem instances. The top row shows the relative error in estimated distances for the Three Point Delta algorithm (TPD) and the Misty Three Point algorithm (MTP). The second row shows the relative error in estimated camera direction for TPD and MTP. The third row shows the relative error in estimated camera translation for TPD and MTP. Lastly, the bottom row presents the relative errors in estimated depth difference (left) and underwater imaging parameters (right) performed by the Three Colors Depth Difference algorithm (TCDD).

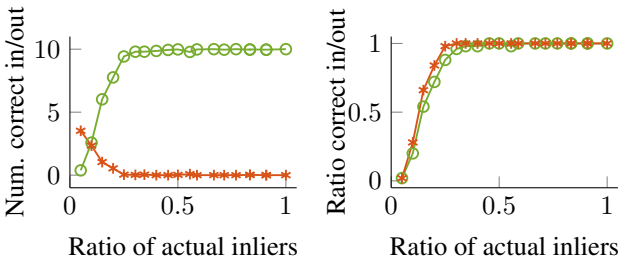

Figure 6. The performance of inlier/outlier classification of the system given a varying number of outliers, while the number of inliers is fixed to 10. In the left figure, the number of classified inliers from the true inlier group (green) and the outlier group(red) is shown. In the right figure, the rate of classifying outliers as outliers (red) and inliers as inliers (green) is plotted. The test for each inlier ratio was repeated 50 times, each of which consisted of 1000 RANSAC iterations, and the plotted lines are the mean values over those tests.

that we can accurately estimate the camera motion. Furthermore, MTP was used to estimate the absolute distance

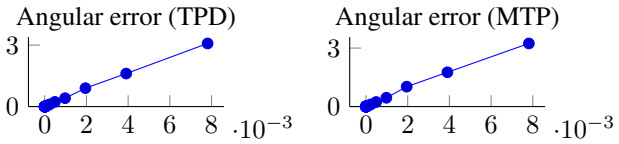

Figure 7. Median of angular error in degrees as a function of noise variance. For each noise level x, 1000 random problem instances were generated, and normal distributed noise with zero mean and x^2 variance was added to the generated points.

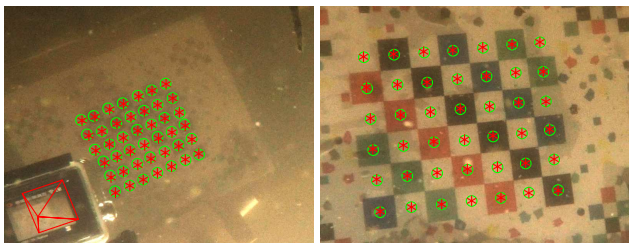

Figure 8. Real evaluation data. Two cameras are fixed in relation to each other, and observe a checkerboard-like planar object from multiple angles and distances, both in air and in water. This figure shows an image from one of the two cameras, taken in water, where ground truth data is plotted as green circles and reprojections are red asterisks. Note that the second camera can be seen in the lower left part of the image, where also its estimated pose is projected and plotted in red.

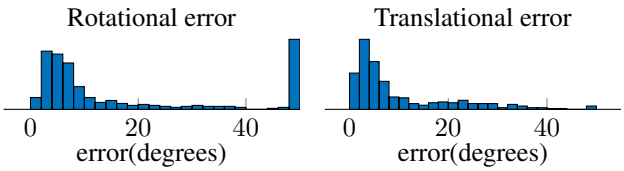

Figure 9. Evaluation of accuracy of Misty Three Point (MTP) relative motion estimates in real experiments, compared to in-air estimated ground truth. The left histogram shows the rotational error in degrees and the right histogram the translational error in degrees, in 1000 repetitions of a 100-iterations RANSAC procedure on data similar to Fig. 8, contaminated to 50% outliers.

given the three projected points and their colors, and TPD provided estimates using the three projected points and the given change in depth. The relative errors of the estimates are presented in Fig 5. The numerical accuracy of TCDD was tested by providing the observed colors of the three points. The estimated difference in depth as well as the underwater imaging parameters are compared to the ground truth in Fig 5.

5.2. Real data

The practical performance of the method was evaluated by fixing two cameras in relation to each other, record videos, and compare the results of the Misty Three Point algorithm (MTP) in water to the best estimate produced by the five-point relative pose algorithm in air. This was achieved

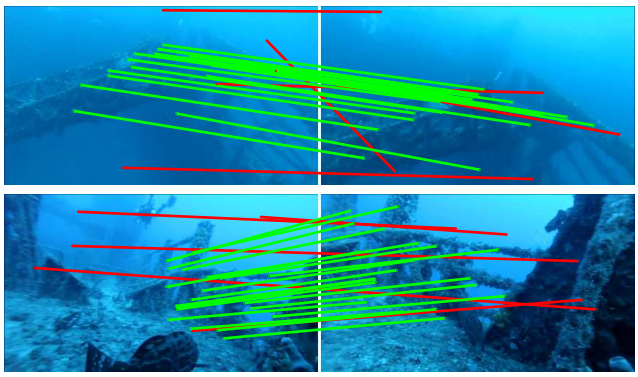

Figure 10. Some examples of images and corresponding points (inliers are green, outliers are red) found using the Misty Three Point algorithm.

by recording videos of a planar calibration pattern at different orientations and distances both in water and in air. First, the relative pose was estimated with high precision on in-air images, to create ground truth. Then, MTP was applied on in-water images (see Fig. 8).

The performance of MTP was evaluated by repeating RANSAC procedures of 100 iterations on data that contain 50% outliers. Then, the estimated relative poses of the second camera were compared to ground truth by measuring the differences in rotation and translation (see Fig. 9). Note that the bins at 50 degrees contain all estimates that produce errors of 50 degrees or larger. This evaluation clearly shows that MTP delivers quantitatively accurate relative pose estimates in a real application.

Further proof-of-concept is provided by qualitatively analyzing the performance of the method applied to an underwater video downloaded from YouTube. The video was captured using a GoPro camera enclosed in a flat port protective underwater housing, with unknown intrinsic calibration. Fig. 1 and Fig. 10 shows that MTP succeeds in finding qualitatively correct corresponding points. Furthermore, Fig. 1 shows that MTP successfully estimates a qualitatively correct sequence of motion.

6. Conclusion

We have proposed a novel method for estimating relative motion given three points and their colors. Using physical models for underwater imaging, we have shown that the depth information that is present in the observed colors can be estimated and used in practice. We also demonstrate that our algorithms perform quantitatively well in the synthetic experiments when compared to ground truth (see Fig 5), when exposed to noise (see Fig 7), and in a RANSAC-framework when exposed to high ratios of outliers (see Fig 6). Furthermore, we have shown that the system performs reasonably well in estimating structure and motion in a real application (see Fig. 9). Lastly, we have shown qualitatively promising results on YouTube video data (Fig. 10).

References

[1] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. Seitz, and R. Szeliski. Building rome in a day. 2009.

[2] A. Agrawal, S. Ramalingam, Y. Taguchi, and V. Chari. A theory of multi-layer flat refractive geometry. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3346–3353. IEEE, 2012.

[3] Y. Bahat and M. Irani. Blind dehazing using internal patch recurrence. In *Computational Photography (ICCP), 2016 IEEE International Conference on*, pages 1–9. IEEE, 2016.

[4] R. Fattal. Single image dehazing. *ACM transactions on graphics (TOG)*, 27(3):72, 2008.

[5] J.-M. Frahm, P. Fite-Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y.-H. Jen, E. Dunn, B. Clipp, S. Lazebnik, et al. Building rome on a cloudless day. 2010.

[6] S. Haner and K. Astrom. Absolute pose for cameras under flat refractive interfaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1428–1436, 2015.

[7] R. M. Haralick, C.-n. Lee, K. Ottenburg, and M. Nölle. Analysis and solutions of the three point perspective pose estimation problem. In *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*, pages 592–598. IEEE, 1991.

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

[9] J. Heinly, J. L. Schönberger, E. Dunn, and J.-M. Frahm. Reconstructing the World* in Six Days *(As Captured by the Yahoo 100 Million Image Dataset). 2015.

[10] J. S. Jaffe. Computer modeling and the design of optimal underwater imaging systems. *Oceanic Engineering, IEEE Journal of*, 15(2):101–111, 1990.

[11] A. Jordt. *Underwater 3D Reconstruction Based on Physical Models for Refraction and Underwater Light Propagation*. PhD thesis, Universitätsbibliothek Kiel, 2013.

[12] A. Jordt-Sedlazeck and R. Koch. Refractive structure-from-motion on underwater images. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2013.

[13] E. Nascimento, M. Campos, and W. Barros. Stereo based structure recovery of underwater scenes from automatically restored images. In *Computer Graphics and Image Processing (SIBGRAPI), 2009 XXII Brazilian Symposium on*, pages 330–337. IEEE, 2009.

[14] J. P. Queiroz-Neto, R. Carceroni, W. Barros, and M. Campos. Underwater stereo. In *Computer Graphics and Image Processing, 2004. Proceedings. 17th Brazilian Symposium on*, pages 170–177. IEEE, 2004.

[15] Y. Y. Schechner and N. Karpel. Clear underwater vision. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–536. IEEE.

[16] Y. Y. Schechner and N. Karpel. Recovery of underwater visibility and structure by polarization analysis. *Oceanic Engineering, IEEE Journal of*, 30(3):570–587, 2005.

[17] A. Sedlazeck and R. Koch. Simulating deep sea underwater images using physical models for light attenuation, scattering, and refraction. 2011.

[18] A. Sedlazeck and R. Koch. *Outdoor and Large-Scale Real-World Scene Analysis: 15th International Workshop on Theoretical Foundations of Computer Vision, Dagstuhl Castle, Germany, June 26 - July 1, 2011. Revised Selected Papers*, chapter Perspective and Non-perspective Camera Models in Underwater Imaging – Overview and Error Analysis, pages 212–242. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[19] N. Snavely, S. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. *ACM Trans. Gr.*, 2006.

[20] H. Stewénius. *Gröbner basis methods for minimal problems in computer vision*. Citeseer, 2005.

[21] Y. Swirski and Y. Y. Schechner. 3deflicker from motion. In *Computational Photography (ICCP), 2013 IEEE International Conference on*, pages 1–9. IEEE, 2013.

[22] T. Treibitz, Y. Schechner, C. Kunz, and H. Singh. Flat refractive geometry. *IEEE transactions on pattern analysis and machine intelligence*, 34(1):51–65, 2012.

[23] G. Young and A. S. Householder. Discussion of a set of points in terms of their mutual distances. *Psychometrika*, 3(1):19–22, 1938.