

PDE Based Dense Depth Estimation for Stereo Fisheye Image Pair and Uncertainty Quantification

Sandesh Athni Hiremath(✉)

Valeo Schalter und Sensoren, Hummendorfer Straße 76, 96317 Kronach-Neuses, Germany
sandesh.athni-hiremath@valeo.com

Abstract. Dense estimation of depth from stereo pinhole image pair is a well known problem and seems to be easily solvable by both classical as well as machine learning algorithms. However, dense depth estimation from stereo fisheye image pair is apparently a more challenging problem. There are two main factors adding to the complexity of the problem: (I) owing to the wider field of view, the fisheye lenses are inherently nonlinear thus making the estimation problem harder. (II) because we want to estimate depth for every pixel coordinate from just two images, i.e. a calibrated stereo pair, the data is statistically insufficient thus greatly complexifying the problem. To alleviate (II) many depth estimation algorithms enlarge their parameter space by incorporating pose parameters and work on image sequence (esp. videos). Here we stick to the stereo camera setting and provide a novel estimation technique and quantify its uncertainty. We use the technique of variational calculus to derive a (noisy) pde for which wellposedness is proved and then solved in the *least square* sense using proximal algorithms. This setting enables us to invoke the standard machinery of nonlinear least squares for generating the covariance estimates. Most state of the art algorithms that are based on deep learning (nonlinear regression) technique fail to say anything about the uncertainty of the outputs which is of paramount importance in safety critical applications for e.g. arising in automotive domain. Lastly, our methodology has the advantage of being massively parallelizable for hardware implementation thereby offering lesser runtime.

Keywords: Stereo fisheye correspondence · Nonlinear optimization · Uncertainty quantification

1 Introduction

Tremendous research in image processing algorithms has given rise to a wide variety of applications across all domains ranging from photography, entertainment, surveillance, medicine, robotics, etc. to name a few. Similarly, even the

© Springer Nature Switzerland AG 2020
O. Gervasi et al. (Eds.): ICCSA 2020, LNCS 12251, pp. 648–663, 2020.
https://doi.org/10.1007/978-3-030-58808-3_47

automotive industry has seen a surge of vision based applications especially for semi or fully automated driving. The range of applications include lane detection, traffic sign recognition, object/obstacle detection, parking assistance and so on. Consequently, cameras are becoming one of the mandatory sensor for an automobile. Due to limitation on the number of installed cameras and the need for incorporating more number of safety features, the choice for a wider angle cameras become quite obvious [7,8]. Owing to this necessity fisheye cameras, whose FOV is larger than 180°, are becoming popular for automotive applications. As a result, many of the computer vision algorithms are being adapted for wide-angle lenses [1–4,9,10,22].

Stereo image correspondence is a well known classical computer vision problem which involves finding the location of the objects of the first image in the second image. For a pinhole cameras this problem simplifies to finding the shift in pixels i.e. disparity between the images. The disparity is then used to calculate the depth of the observed objects, thereby enabling a dense 3D reconstruction. Many well know algorithms, both classical as well as deep learning based approaches, exists for this problem [5,6,14,17,23] and some of them serve as a standard in industry. However, the same correspondence problem for fisheye images is still a open problem. The main challenge when dealing with fisheye image is the distortion introduced by the lens (see [1]). This distortion is modeled by a nonlinear camera projection model. A simple workaround is to use the camera projection model to remove the distortion and produce a rectified stereo image pair for which pinhole stereo algorithms can be applied. This approach was followed in [20] which heavily relied of the stable disparity outputs of the standard stereo matching algorithms such as [5,6].

However, there is a growing demand for algorithms that directly work on raw distorted images mainly due to growth in automotive, robotics and drone applications. In [16], the authors used conventional stereo matching algorithms using directly a nonlinear fisheye projection model and in [13] PDE based technique was used to find image correspondences. Analogous to [14], in [19] the authors have used variational methods to directly match fisheye images obtained at different poses by locally linearizing the error term. In contrast to these classical algorithms, authors in [15] adapted deep learning techniques, where they modified/adapted the existing network for pinhole models and trained it on fisheye images. One of the main aspect lacking in all these methods is that they fail to quantify the uncertainty of their estimates and (one of the) goal of this paper is to fill this gap by providing a probabilistic modeling of the problem and to quantify the uncertainty of the estimates.

The paper is organized in the following manner, first we formulate the problem and propose a model in Sect. 2, in Sect. 3 we provide a probabilistic interpretation of the model and propose a method for estimating the variance, in Sect. 4 we propose a numerical method for solving the model and provide results. Finally in Sect. 5 we discuss the reults and make some concluding remarks.

2 Modelling

The problem we want to solve is the following: given a pair of images I_L and I_R obtained from two calibrated fisheye cameras forming a stereo setup, estimate the depth densely i.e. for every pixel of the image grid.

2.1 Problem Formulation

Mathematically, this can be written in the following precise way- given two calibrated fisheye images $I_L, I_R : \mathfrak{D} \rightarrow \mathfrak{R}$, with compact domain $\mathfrak{D} \subset \mathbb{R}^2$ and range $\mathfrak{R} \subset \mathbb{R}^3$, find the mapping $\lambda : \mathfrak{D} \rightarrow \mathbb{R}^+$ that indicates the depth of objects represented in the image I_L(or I_R)[1].

To solve this problem we need to appeal to the underlying epipolar constraints. According to that, if the baseline of the stereo camera is sufficiently wide, images I_L and I_R will have significant overlap and could be viewed as shifted versions of each other. However, for fisheye cameras, unlike in pinhole cameras, the shift is along a curve[2] instead of along a line. The curve is obtained by projecting the curve on the right camera (resp. left camera) surface, traced by a light ray incident on the left camera (resp. right cam), onto the right image (resp. left image) plane. For fisheye and other omnidirectional lenses, images are distorted more on the edges as compared to the middle. Based on the coordinate position of the left image (resp. right image) the curves traced on I_R is of varied degree of curvature, this again adds to the complexity.

Different models for fisheye cameras have been proposed [1,11,21] which have varied degree of accuracy and simplicity. For our work, we use the *enhanced unified camera model* (EUCM) [12] mainly because of its simplicity and reasonable accuracy.

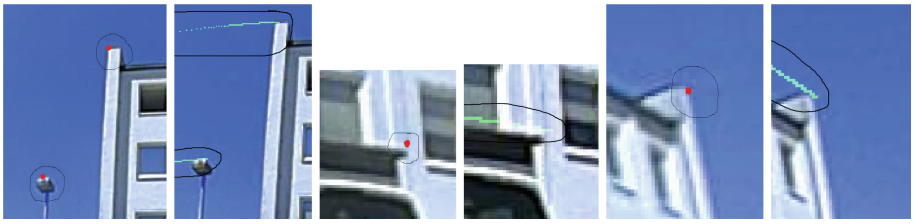
In Figs. 1a–c we illustrate the validity of the EUCM projection model. Firstly, for a given pair of fisheye stereo images of a scene, a set of prominent matching features are selected. The ray passing through a feature point of left image would trace a curve (as a function of depth) on the right image. For the correct calibration parameters, the curve passes through the exact same feature point corresponding to the left image. The point where the curve hits the matching feature point represents the true depth value. The feature points are marked in black dots in left image and the curves are marked in cyan on the right image. The curves pass through the feature points (nearly) exactly. Thus the correctness of projection model and the calibration of the involved parameters is validated.

2.2 Optimization Model

As per the problem formulation Sect. 2.1, the aim is to estimate a positive valued scalar function λ. To this end, we adopt the framework of 'calculus of variations'.

[1] Here we always associate λ as a depth mapping for the left image I_L.

[2] Based on vertical or horizontal stereo setup, the curve extends either vertically or horizontally.



(a) Feature point 1 and 2 (b) Feature point 3 (c) Feature point 4

Fig. 1. The Figs. 1a–c show the magnification of the feature points (shown as red dot) of the left image and curve (cyan color) that passes through those respective features on the right image. The pairs are the subimages of the corresponding fisheye stereo image pair. (Color figure online)

Before we proceed, let us make some basic assumptions on the functions and their spaces. We assume that I_L, I_R are the elements of the Hilbert space $L^2(\mathfrak{D})$ and we expect λ also be an element of $L^2(\mathfrak{D})$.

In order to invoke the tools variational calculus, we first need to define a loss functional whose minimizing element represents the solution to the problem reasonably well.

$$E[\lambda(u,v)] = \frac{1}{2}\int_{\mathfrak{D}} \overbrace{\|I_L(u,v) - I_R(\hat{u}(\lambda),\hat{v}(\lambda))\|^2_{L^2}}^{\text{Data fidelity term}} + \overbrace{s^2(\lambda,\partial_u\lambda,\partial_v\lambda)}^{\text{regularizer term}}\, d\mu \qquad (1)$$

1. $(u,v) \in \mathfrak{D} \subset \mathbb{R}^2$
2. $\lambda : \mathfrak{D} \to \mathbb{R}$ is the depth map which needs to be estimated
3. $I_L : \mathfrak{D} \to \mathbb{R}$ and $I_R : \mathfrak{D} \to \mathbb{R}$ are the left and right fisheye images forming a stereo pair.
4. $(\hat{u},\hat{v}) \in \mathfrak{D}$ is the pixel coordinate of the right image for the corresponding pixel coordinate $(u,v) \in \mathfrak{D}$ of the left image. More precisely we have that $(u,v) \mapsto \hat{u}(\lambda(u,v))$ and $(u,v) \mapsto \hat{v}(\lambda(u,v))$
5. $(u,v) \mapsto s(u,v)$ is the magnitude of the normal vector to the tangent plane, at the position (u,v) on the surface generated by the 3D point $Z(\lambda(u,v))$.

The **data fidelity** term is based on the brightness constancy assumption and is required to minimize the error in reconstruction of the first image after sampling from the second image.

The **regularizer** term is required to ensure structural consistency as well as to ensure solvability of the minimization problem and the candidacy the solution in the appropriate function space. The intuitive meaning of the term used here is that $(u,v) \mapsto s(u,v)$ represents the infinitesimal area of the 3D surface generated by the mapping

$$(u,v) \mapsto Z(\lambda(u,v)) := \lambda(u,v)\, X(u,v).$$

Here Z represents the 3d world point which can also be seen as a function of the depth map $(u,v) \mapsto \lambda(u,v)$. More precisely, we also have that $\lambda \mapsto Z(\lambda) := \lambda X$.

Note that, since $X(u,v)$ is obtained after unprojecting the image coordinates (using the projection model), it is of unit norm and remains constant for a given set of camera intrinsics. Consequently, Z varies as per the depth map $\lambda(u,v)$. Based on this, the area of the surface spanned by Z can be computed in the following way:

$$S = \iint s(u,v)\,du\,dv := \iint |Z_u \times Z_v|\,du\,dv$$

where, Z_u and Z_v are the partial derivatives of Z w.r.t u and v respectively. Altogether, we have that

$$S = \iint s(u,v)\,du\,dv := \iint |Z_u \times Z_v|\,du\,dv := \iint |a\lambda_u\lambda + b\lambda\lambda_v + c\lambda^2|\,du\,dv.$$

where, $(u,v) \mapsto a(u,v) \in \mathbb{R}^3$, $(u,v) \mapsto b(u,v) \in \mathbb{R}^3$, and $(u,v) \mapsto c(u,v) \in \mathbb{R}^3$ are constant vector functions given as

$$a := X \times X_v = (a_1, a_2, a_3)^T,\ b := X_u \times X = (b_1, b_2, b_3)^T,\ c := X_u \times X_v = (c_1, c_2, c_3)^T.$$

From the above formula it is clear that the regularizer term s^2 in (1) is directly inducing a smoothness constraint on the depth map.
To estimate the depth map for either one of the stereo fisheye image pair (I_L, I_R), we need to solve the following variational problem:

$$\hat{\lambda}(u,v) = \underset{\lambda \in L^2(\mathfrak{D})}{\operatorname{argmin}}\ E[\lambda(u,v)]. \tag{2}$$

A necessary condition for $\hat{\lambda}$ to be the minimizer of $E[\lambda]$ is that it must satisfy the corresponding Euler-Lagrange equation which is given as:

$$\Big(I_L(u,v) - I_R(\hat{u}(\hat{\lambda}), \hat{v}(\hat{\lambda}))\Big)\Big(\nabla I_R|_{\hat{\lambda}} \cdot \nabla \hat{\mathbf{u}}|_{\hat{\lambda}}\Big) - \partial_\lambda s^2 + \nabla \cdot (\nabla_{\lambda_u, \lambda_v} s^2) = 0 \tag{3}$$

The partial derivatives of s^2 w.r.t λ are computed in the following way: let

$$s(\lambda, \partial_u\lambda, \partial_v\lambda) := |Z_u \times Z_v| = |a\lambda_u\lambda + b\lambda\lambda_v + c\lambda^2|, \tag{4}$$

then

$$\begin{aligned}
\partial_\lambda s^2 &= 2\Big[\sum_{k=1}^{3} \lambda(\lambda_u a_k + \lambda_v b_k + \lambda c_k)^2 + \sum_{k=1}^{3} \lambda^2 c_k(\lambda_u a_k + \lambda_v b_k + \lambda c_k)\Big], \\
\partial_{\lambda_u} s^2 &= 2\Big[\lambda^2\big(\lambda_u \|a\|_2^2 + \lambda_v (a,b)_2 + \lambda (c,a)_2\big)\Big], \\
\partial_{\lambda_v} s^2 &= 2\Big[\lambda^2\big(\lambda_u (a,b)_2 + \lambda_v \|b\|_2^2 + \lambda (c,b)_2\big)\Big].
\end{aligned}$$

Next we make an important simplification, namely we want to convexify the regularizer term. Note that, from Eq. (4) it is clear that the mapping $\lambda \mapsto s^2(\lambda, \cdot, \cdot)$ is a quadric (4th-order) polynomial, consequently its first order derivatives will be at most a cubic polynomial which in general is non-convex. In order to account

for this we simply define a new mapping $\lambda \mapsto \hat{s}(\lambda, \cdot, \cdot) := \frac{1}{\lambda} s(\lambda, \cdot, \cdot)$ and take $\hat{s}$ as the regularizer term. As per this, we have the following modified loss functional

$$\hat{E}[\lambda(u,v)] = \tfrac{1}{2} \int_{\mathfrak{D}} \overbrace{\|I_L(u,v) - I_R(\hat{u}(\lambda), \hat{v}(\lambda))\|_{L^2}^2}^{\text{Data fidelity term}} + \overbrace{\hat{s}^2(\lambda, \partial_u\lambda, \partial_v\lambda)}^{\text{convexified regularizer term}} \; d\mu. \quad (5)$$

Its corresponding Euler-Lagrange equation reads as

$$\Big(I_L(u,v) - I_R(\hat{u}(\hat{\lambda}), \hat{v}(\hat{\lambda}))\Big)\Big(\nabla I_R|_{\hat{\lambda}} \cdot \nabla \hat{\mathbf{u}}|_{\hat{\lambda}}\Big) - \partial_\lambda \hat{s}^2 + \nabla \cdot (\nabla_{\lambda_u, \lambda_v} \hat{s}^2) = 0, \quad (6)$$

where,

$$\begin{aligned}
\hat{s}(\lambda, \partial_u\lambda, \partial_v\lambda) &:= \frac{1}{\lambda}|Z_u \times Z_v| = |a\lambda_u + b\lambda_v + c\lambda| \\
\partial_\lambda \hat{s}^2 &= 2\big(\lambda_u(a,c) + \lambda_v(b,c) + \lambda\|c\|_2^2\big), \\
\partial_{\lambda_u} \hat{s}^2 &= 2\big(\lambda_u\|a\|_2^2 + \lambda_v(a,b)_2 + \lambda(c,a)_2\big), \\
\partial_{\lambda_v} \hat{s}^2 &= 2\big(\lambda_u(a,b)_2 + \lambda_v\|b\|_2^2 + \lambda(c,b)_2\big).
\end{aligned}$$

Equation (6) can be abstractly written in the following form:

$$A(u,v)\lambda = f(\lambda), \quad (7)$$

where,

1. the mapping $(u,v) \mapsto A(u,v) \in L(D(A), L^2(\mathfrak{D}))$ with

$$A(u,v) := \nabla \cdot \big(K_1(u,v)\,\nabla\lambda\big) + K_2(u,v) \cdot \nabla\lambda - r(u,v)\lambda,$$

where

$$\begin{aligned}
K_1(u,v) &:= \kappa_1 \begin{pmatrix} p_1 & p_2 \\ p_2 & p_4 \end{pmatrix} := \kappa_1 \begin{pmatrix} \|a\|_2^2 & (a,b)_2 \\ (a,b)_2 & \|b\|_2^2 \end{pmatrix} \\
K_2(u,v) &:= \kappa_2 \begin{pmatrix} p_3 \\ p_5 \end{pmatrix} := \kappa_2 \begin{pmatrix} (a,c)_2 \\ (b,c)_2 \end{pmatrix} \\
r(u,v) &:= q(u,v) + \kappa_2 \begin{pmatrix} p_3 \\ p_5 \end{pmatrix} \cdot \nabla \\
q(u,v) &:= \kappa_3\|c\|_2^2
\end{aligned}$$

2. the mapping $\lambda \mapsto f(\lambda) \in L^2(\mathfrak{D})$ with

$$f(\lambda) := -\Big(I_L(u,v) - I_R(\hat{u}(\lambda), \hat{v}(\lambda))\Big)\Big(\nabla I_R|_{\hat{\lambda}} \cdot \nabla \hat{\mathbf{u}}|_{\hat{\lambda}}\Big) - \kappa_4\,(\lambda\nabla \cdot K_2(u,v))$$

3. $\kappa := (\kappa_1, \kappa_2, \kappa_3, \kappa_4)$ is a vector of model parameters and is a positive constant.

In order to solve (7) we still need to enforce boundary condition for which we make the obvious choice of Neumann boundary condition.

Theorem 1. *Let the images I_L, $I_R \in H^1(\mathfrak{D}) \cap L^\infty(\mathfrak{D})$, the functions $a(u,v)$, $b(u,v)$, $c(u,v) \in C^2(\mathfrak{D}) \cap L^\infty(\mathfrak{D})$, the coordinate functions $\hat{u}$ and $\hat{v} \in C^2(\mathfrak{D}) \cap L^\infty(\mathfrak{D})$. Additionally, let $K_1(u,v)$ is uniformly positive definite for each $u, v \in \mathfrak{D}$, i.e. $\xi^T K_1 \xi \geq \theta|\xi|^2$, $\theta > 0$, $\forall \xi \in \mathbb{R}^2, (u,v) \in \mathfrak{D}$ and $q(u,v) \geq q_{min} > 0$. Then the pde given in (7) has a unique solution $\lambda \in H^1(\mathfrak{D})$.*

Proof. Consider the bilinear form $B[\phi,\psi] := \int_{\mathfrak{D}} K_1 \nabla\phi \cdot \nabla\psi + \int_{\mathfrak{D}} q\phi\psi$

$$B[\phi,\psi] = \int_{\mathfrak{D}} p_1 \partial_u\phi\partial_u\psi + 2(p_2\partial_v\phi\partial_u\psi) + p_4\partial_v\phi\partial_v\psi + \int_{\mathfrak{D}} q\psi\phi$$
$$\Rightarrow |B[\phi,\psi]| \leq C\|\phi\|_{H^1}\|\psi\|_{H^1}$$

$$B[\phi,\phi] = \int_{\mathfrak{D}} (\partial_u\phi a, \partial_u\phi a)_2 + 2(\partial_v\phi a, \partial_u\phi b)_2 + (\partial_v\phi b, \partial_v\phi b)_2 + \int_{\mathfrak{D}} q\phi\phi$$
$$\geq \theta\|\phi\|^2_{H^1}$$

Thus by Lax-Milgram we have an unique solution to $A\lambda = g$ in $H^1(\lambda)$ for $g := f(\lambda) \in L^2(\mathfrak{D})$. This implies the resolvent operator $A^{-1} : L^2 \to H^1$ maps every $g \in L^2$ to a corresponding $\lambda \in H^1$.
Since $\|\lambda\|^2_{H^1} \leq \frac{1}{\theta} B[\lambda,\lambda] \leq \frac{1}{\theta}\|g\|_2\|\lambda\|_{H^1}$ we get that $\|A^{-1}g\|_\cdot = \|\lambda\|_\cdot \leq \|\lambda\|_{H^1} \leq \frac{1}{\theta}\|g\|_{L^2}$, we get that

$$\|A^{-1}g\|_{H^1} \leq \frac{1}{\theta}\|g\|_{L^2}, \quad \text{as well as} \quad \|A^{-1}g\|_{L^2} \leq \frac{1}{\theta}\|g\|_{L^2}.$$

Hence, both $A^{-1} : L^2 \to H^1$ and $A^{-1} : L^2 \to L^2$ are continuous linear operators. Howerver, since $H^1 \hookrightarrow\hookrightarrow L^2$ the operator $A^{-1} : L^2 \to L^2$ is compact.
Now the weak formulation of the original problem is equivalent to finding a fixed point of $T\lambda := A^{-1}f(\lambda)$. Following as above, we have $\|T\lambda\|_\cdot = \|A^{-1}f(\lambda)\|_\cdot \leq \frac{1}{\theta}\|f\|_{L^2} \leq \frac{1}{\theta}\|f\|_{L^\infty}$. Thus maps bounded sets in L^2 to bounded sets in H^1 and since $H^1 \hookrightarrow\hookrightarrow L^2$ we get that $T : L^2 \to L^2$ is a compact operator. To show that $T : L^2 \to L^2$ is also continuous, consider a sequence $(\lambda_m)_{m\in\mathbb{N}}$ such that $\lambda_m \to \lambda$ in $L^2(\mathfrak{D})$, then there exists a subsequence $(\lambda_{m_k})_{k\in\mathbb{N}}$ such that $\lambda_{m_k}(\cdot) \to \lambda(\cdot)$ a.e. in $\mathfrak{D}$. Thus, by continuity of A^{-1} we have that

$$\|T\lambda_{m_k} - T\lambda\|_{L^2} = \|A^{-1}f(\lambda_{m_k}) - A^{-1}f(\lambda)\|_{L^2} = \|A^{-1}(f(\lambda_{m_k}) - f(\lambda))\|_{L^2}$$
$$\leq \frac{1}{\theta}\|f(\lambda_{m_k}) - f(\lambda)\|_{L^2}$$

Since, $f \in C^2$, $\lambda_{m_k} \to \lambda$ implies $\|f(\lambda_{m_k}) - f(\lambda)\|_{L^2} \to 0$ as $k \to \infty$. Thus we also get that $\|T\lambda_{m_k} - T\lambda\|_{L^2} \to 0$ as $k \to \infty$. Finally, we have that $T : L^2 \to L^2$ is compact and continuous in L^2 with $T(H^1) \subset H^1$ thus by Shauder's fixed point theorem we get that $\exists\lambda \in L^2$ such that $\lambda = T\lambda$ in $L^2(\mathfrak{D})$. Moreover, since $\|T\lambda\|_{H^1} = \|T\lambda\|_{H^1} \leq C < \infty$ implies that $\lambda \in H^1(\mathfrak{D})$.

Due to the boundedness of the $\mathfrak{D} \subset \mathbb{R}^2$, I_L, I_R clearly belong to $L^2(\mathfrak{D})$. Because of the continuity and boundedness of the EUCM projection model, the

coordinate functions $\hat{u}, \hat{v}$ also smooth and bounded. Similarly, even the cross product functions a, b, c can be verified to be smooth and bounded. To ensure uniform ellipticity of the diffusion coefficient and edge aware diffusion we modify $K_1(u, v)$ as:

$$K_1(u,v) := \kappa_1 \begin{pmatrix} \exp\left[-\kappa_5|\partial_u I_L| + \kappa_6(1+p_1)\right] & \exp\left[-\frac{\kappa_5}{2}(|\partial_u I_L| + |\partial_v I_L|) + \kappa_6(1+p_2)\right] \\ \exp\left[-\frac{\kappa_5}{2}(|\partial_u I_L| + |\partial_v I_L|) + \kappa_6(1+p_2)\right] & \exp\left[-\kappa_5|\partial_v I_L| + \kappa_6(1+p_4)\right] \end{pmatrix} \tag{8}$$

Since $|\partial_x I_L|$ and $|\partial_y I_L|$ are uniformly bounded, K_1 is uniformly elliptic if and only if $p_1 + p_4 > 2p_2$ i.e. $\|a\|^2 + \|b\|^2 - (a, b)_2 > 0$ i.e. $\|a - b\|^2 > 0$. If necessary, a sufficiently small constant is added to the diagonal elements, (especially where $a(u, v) = b(u, v)$ or $a = b = 0$), of the K_1 in order to make it strictly positive definite.

3 Uncertainty Quantification

3.1 Noise Model

The necessary condition for finding the minimum of the Functional (5) is to solve the Eq. (7). In order to consider the errors induced in the image formation we incorporate noise in the model which takes the following form:

$$A(u, v)\lambda = f(\lambda) + W, \tag{9}$$

where $W \sim N(0, \Sigma(\lambda))$ is a $L^2(\mathfrak{D})$-valued Gaussian random variable with covariance operator $\Sigma \in L(L^2(\mathfrak{D}); L^2(\mathfrak{D}))$. The noise term depends on the depth function which facilitates positivity of the solution. It is important to note here that, even though we have introduced the noise at the level of Euler-Lagrange equation (7), one may also, equivalently, introduce at the level of functionals (5). Intuitively, what this means is that, the data fidelity term reads as $\|I_L(u, v) - I_R(u'(\lambda), v'(\lambda)) + W\|^2_{L^2}$, where $W \sim N(0, \Sigma(\lambda))$ is a $L^2(\mathfrak{D})$ valued Gaussian random variable.

The noisy model is now solved in the L^2 sense i.e.

$$\underset{\lambda \in L^2(\mathfrak{D})}{\operatorname{argmin}} \frac{1}{2}\|A\lambda - f(\lambda)\|^2_{\Sigma^{\frac{1}{2}}(L^2)}. \tag{10}$$

We note here that the solution to a least squares problem is given by solving the normal equation.

Theorem 2 (Least squares solution)**.** *Let H_1 and H_2 be two Hilbert spaces, let $A : H_1 \to H_2$ be a bounded linear operator. Then the least squares solution to $Ax = y$ for $x \in H_1$ and $y \in H_2$ is given as $\hat{x} = \underset{x \in H_1}{argmin}\ \frac{1}{2}\|Ax - y\|^2_{H_2}$ and is obtained by solving the normal equation*

$$A^*Ax = A^*y \quad i.e \quad \hat{x} := (A^*A)^\dagger A^*y,$$

where $(A^*A)^\dagger$ denotes the pseudo inverse of A^*A.

This is a nonlinear least squares problem which we solve following the techniques provided in the next section. In order to numerically solve (10) we invoke the machinery of convex optimization esp. techniques involving the use of proximal operators. Additionally, it can be supplemented by constraints such as positivity, smoothness, shearlet, wavelet and so on. To facilitate this we consider the following generic minimization formulation:

$$\underset{x,y}{\operatorname{argmin}} \frac{1}{2}\|Ax - f\|_2^2 + \iota_S(y) \quad \text{s.t.} \quad Qx = y \tag{11}$$

Letting ι_S be the indicator function of the set S and Q a bounded linear operator. We now mention a primal-dual algorithm that is used to solve the minimization problems of the type (11).

Algorithm 1: PDHGMx

Data: $\kappa > 0$, τ_x, $\tau_p > 0$, $\vartheta \in (0, 1]$, x^0, p^0

1 $\bar{x}^0 := x^0$
2 $\tilde{x}^0 := x^0$
3 **for** $n = 0, 1 \ldots$ **do**
4 $\quad p^{n+1} = p^n + \tau_p Q\bar{x}^n - \tau_p P_S\left(\frac{1}{\tau_p}p^n + Q\bar{x}^n\right)$
5 $\quad \tilde{x}^{n+1} = x^n - \tau_x Q^* p^{n+1}$
6 $\quad x^{n+1} = (A^*A + \frac{1}{\tau_x})^{-1}(A^*f + \frac{1}{\tau_x}\tilde{x}^{n+1})$
7 $\quad \bar{x}^{n+1} = x^{n+1} + \vartheta(x^{n+1} - x^n)$
8 **end**

3.2 Covariance Computation

Based on the Algorithm 1 we can compute the uncertainty of the estimate x^{n+1} in the following way. Let $\hat{A} = [(A, 0), (0, \frac{1}{\sqrt{\tau_x}})]$ and $\hat{f} = [f, \frac{1}{\sqrt{\tau_x}}\tilde{x}^{n+1}]$. Furthermore, let $\hat{f} \sim N(\mu_{\hat{f}}, \Sigma_{\hat{f}})$, then line-6 can be seen as a solution to the following problem

$$x^{n+1} = \underset{x}{\operatorname{argmin}} \frac{1}{2}\|\hat{A}x - \hat{f}\|^2_{\Sigma_{\hat{f}}^{\frac{1}{2}}}.$$

The solution to this is given as $x^{n+1} \in N(\mu_x, \Sigma_x)$ with

$$\mu_x = (\hat{A}^*\Sigma_f^{-1}\hat{A})^{-1}\hat{A}^*\Sigma_{\hat{f}}^{-1}\mu_f \text{ and } \Sigma_x = (\hat{A}^*\Sigma_{\hat{f}}^{-1}\hat{A})^{-1}.$$

For a special case when f and $\tilde{x}^{n+1}$ are independent of each other, Σ_f and Σ_f^{-1} can be written as

$$\Sigma_f = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \text{ and } \Sigma_f^{-1} = \begin{bmatrix} \Sigma_1^{-1} & 0 \\ 0 & \Sigma_2^{-1} \end{bmatrix}$$

where $f \sim N(\mu_1, \Sigma_1)$ and $\tilde{x}^{n+1} \sim N(\mu_2, \Sigma_2)$. Correspondingly, we have that

$$\hat{A}^*\Sigma_f^{-1}\hat{A} = A^*\Sigma_1^{-1}A + \frac{1}{\tau_x}\Sigma_2^{-1}, \quad \hat{A}^*\Sigma_f^{-1}\mu_f = A^*\Sigma_1^{-1}\mu_1 + \frac{1}{\tau_x}\Sigma_2^{-1}\mu_2$$

and

$$\Sigma_x = \left(A^* \Sigma_1^{-1} A + \frac{1}{\tau_x} \Sigma_2^{-1}\right)^{-1}. \tag{12}$$

4 Implementation

As mentioned above (Sect. 2) we are interested in estimating the depth map corresponding to an raw image obtained by fisheye cameras in a stereo setup. Firstly, the intrinsic camera parameters K_L and K_R are determined and kept fixed. Similarly, the relative transformation R and T between the two cameras are also determined and kept fixed. The operators R and T is used to transform the origin of left camera into the origin of the right camera as explained above in Sect. 2. The images I_L and I_R obtained are converted to grayscale images and are normalized to $[0, 1]$. The raw images are of resolution 800×1200 which are cropped to 512×1024 in order to consider only the main overlapping regions of the images. Finally, the 512×1024 images are downscaled to 128×256 and fed for optimization. The obtained depth map of size 125×512 which is then upscaled to the actual size. Since (10) is nonlinear we solve it locally and iteratively, i.e.

$$\lambda_{m+1} = \underset{\lambda \in L^2(\mathfrak{D})}{\operatorname{argmin}} \frac{1}{2} \|A\lambda - f(\lambda_m)\|^2_{\Sigma^{\frac{1}{2}}(L^2)}. \tag{13}$$

The constraints $a > 0, b > 0, c > 0$ and $a \neq b$, mentioned below formula (8), were numerically verified for the parameters in Table 1. This ensures uniform ellipticity of K_1 as a result of which we get existence of a weak solution. Consequently, the least square solution is equivalent to the weak solution. Moreover, to ensure positivity of the solution, we enforce box-constraints i.e. $0 < \lambda_{min} \leq \lambda \leq \lambda_{max}$ with $\lambda_{min} := .5$ and $\lambda_{max} := 100$. By setting $Q := I$ and $P_S := P_{[\lambda_{min}, \lambda_{max}]}$ in Algorithm 1 we get the Algorithm 2.

For evaluating I_R at non integer pixel locations $(\hat{u}, \hat{v})$ simple bilinear interpolation was used. For solving the resulting matrix equations, algebraic multigrid solvers were used [18]. For the covariance estimate we use the formula (12) with

$$\Sigma_1 := \left(\frac{\lambda}{(1+\lambda)} (\nabla I_R|_\lambda \cdot \nabla \hat{\mathbf{u}}|_\lambda)\right)^2 \text{ and } \Sigma_2 := I. \tag{14}$$

Since we are solving the quadratic problem (Eq. (13)) locally, numerical values of Σ_1 are obtained by evaluating (14) at λ_m^n. Based on this we have that $\Sigma_{\lambda_m^{n+1}} = \left(A^*(\Sigma_1(\lambda_m^n))^{-1} A + \frac{1}{\tau_x}\right)^{-1}$. As $\lambda_m^n \to \hat{\lambda}$ in $L^2(\mathfrak{D})$ we have $\Sigma_1(\lambda_m^n) \to \Sigma_1(\hat{\lambda})$, consequently we also get that $\Sigma_{\lambda_m^n} \to \Sigma_{\hat{\lambda}}$. Since convariance computation involves matrix inversion it is a relatively expensive operation. In order to make the computation practically tractable, we make the following simplification: (i) during covariance computation we downscale the image to the

Algorithm 2: PDHGMx

Data: $\kappa > 0$, τ_x, $\tau_p > 0$, $\vartheta \in (0, 1]$, λ^0, $p^0 = 0$, tol $= 10^{-15}$

Initialization: $\bar{\lambda}_0^0 := \lambda^0$, $\tilde{\lambda}_0^0 := \lambda^0$, $\Sigma_{\lambda_0^0} := I$

for $m = 0, 1 \ldots$ **do**

  $f_m := f(\bar{\lambda}_m^0)$

  for $n = 0, 1 \ldots$ **do**

    $p_m^{n+1} = p_m^n + \tau_p \bar{\lambda}_m^n - \tau_p P_S\left(\frac{1}{\tau_p} p_m^n + \bar{\lambda}_m^n\right)$

    $\tilde{\lambda}_m^{n+1} = \lambda_m^n - \tau_x p_m^{n+1}$

    $\lambda_m^{n+1} = (A^* A + \frac{1}{\tau_x})^{-1}(A^* f_m + \frac{1}{\tau_x}\tilde{\lambda}^{n+1})$

    $\bar{\lambda}_m^{n+1} = \lambda_m^{n+1} + \vartheta(\lambda_m^{n+1} - \lambda_m^n)$

    $\Sigma_{\lambda_m^{n+1}} = \left(A^*(\Sigma_1(\lambda_m^n))^{-1} A + \frac{1}{\tau_x}\right)^{-1}$

  end

  $\bar{\lambda}_{m+1}^0 = \bar{\lambda}_m^{n+1}$

  $\tilde{\lambda}_{m+1}^0 = \bar{\lambda}_m^{n+1}$

  if $\|\bar{\lambda}_m^{n+1} - \lambda_m^{n+1}\| < tol$

    break;

  endif

end

Table 1. Simulation parameters

	Model parameters						Numerical parameters					
	κ_1	κ_2	κ_3	κ_4	κ_5	κ_6	ϑ	τ_x	τ_p	m	n	Tol
Scene1	.01, .001	1	1	.001	.05	.1	1	1	.001	35	1	10^{-15}
Scene2	.01, .005	1	1	.001	.05	.1	1	1	.001	35	1	10^{-15}
Scene3	.01	1	1	.1	.05	.1	1	1	.001	25	1	10^{-15}
Scene4	.01	1	1	.1	.05	.1	1	1	.001	25	1	10^{-15}

size 32×64, and (ii) compute only the variance for these pixels. The obtained variance image is then upscaled to the original resolution of 512×1024.

Due to the nonlinearity of the problem the choice of initial condition is highly important. In order to generate a good initial estimate one may use any available algorithms ranging from naive exhaustive global search based method or SGM mehtod or an estimate generated from DNN. We take the output of the FisheyeDistanceNet [15] and use the semantic segmentation output to assign max values to the regions of the image that represent sky. It is specially important for outdoor scenes where sky occupies majority of the scene. This strategy motivates one to supplement an offline algorithm, such as multi-task (trained to infer semantic and depth values) DNN, with an online estimation algorithm. The combination provides two fold advantage: (i) on the one hand it provides a more realistic result by improving the initial estimate based on the actual data and (ii) on the other hand improves safety of the system by rectifying grossly

erroneous DNN outputs. We use of this philosophy to estimate depths for different outdoor scenes. Their results are as shown below in Fig. 2. Figures 2b–q show the estimated depths. The use of semantically segmented Figs. 2a–p has enabled easy identification of the sky region resulting in a much clear depth output. Figures 3a and 3b depict the convergence of the numerical algorithm for all four scenes considered. Scene3 and Scene4 are taken from a video, for which we used more than two images to obtain image correspondences. Here we have used 6 pairs, the spikes in the convergence plot indicative of the use of a new neighboring frame as the sampling image. This was done purely to indicate the plausible improvement by incorporating multiple views that serves to improves statistical sufficiency of the data. This improvement was validated by comparing the estimate with the ground-truth generated by the Valeodyne sensors.

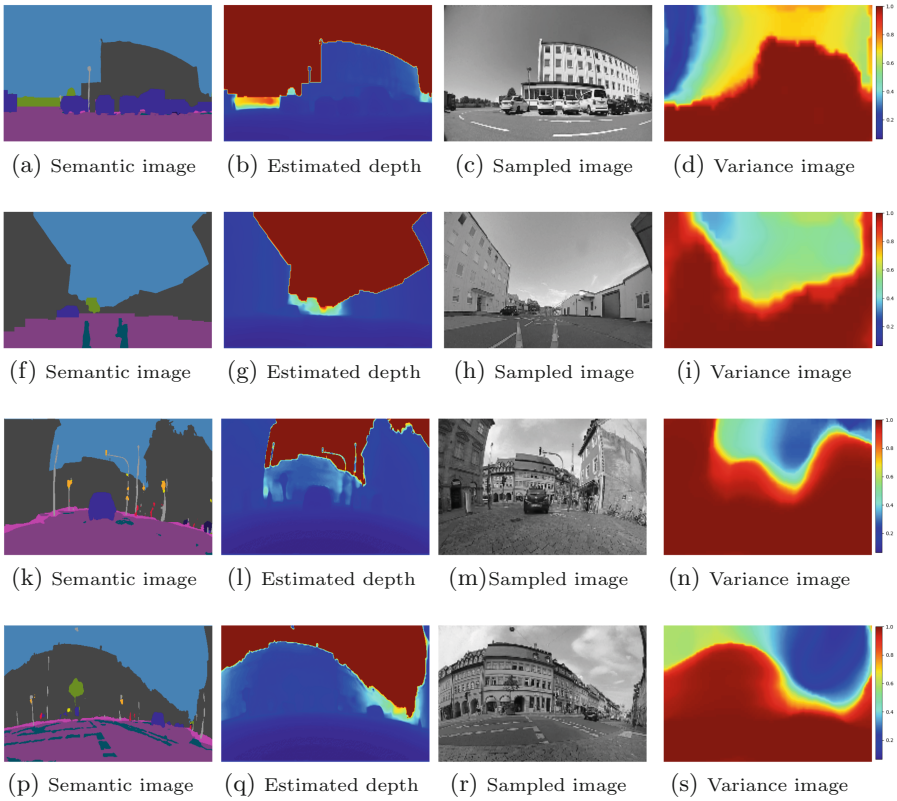


Fig. 2. Simulation results for 4 different scenes: Scene1-Scene4. Each row depicts the result of one particular scene. The left most column is the semantic segmentation of the scene taken as an input for sky identification. The second (from left) column is the estimated depth image. Third (from left) column is the reconstructed target image generated by sampling the right image. The rightmost column is the variance of the estimated depth.

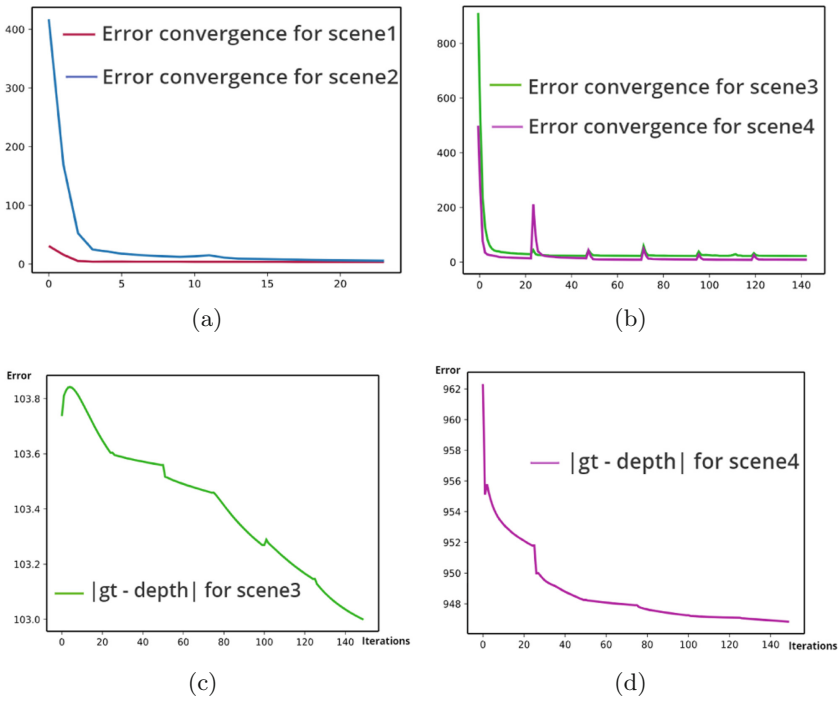
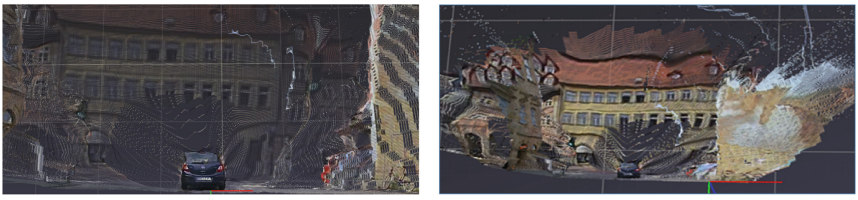
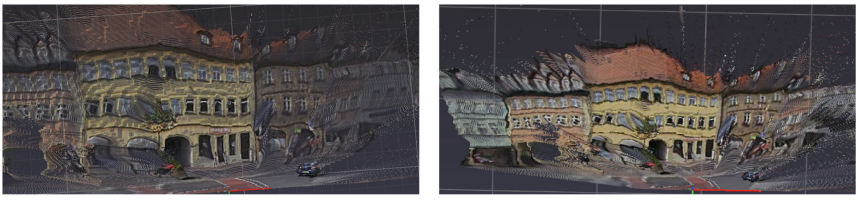


Fig. 3. Plots of convergence errors and ground truth errors. Figure 3a depict the convergence of the Algorithm 2 for Scene1 and Scene2. Figure 3b depict the convergence result for Scene3 and Scene4. Figures 3c and 3d shows the difference of the predicted depth with the available ground truth (gt) values.



(a) Point cloud of scene3 obtained from its estimated depth image



(b) Point cloud of scene4 obtained from its estimated depth image

Fig. 4. Reconstructed 3d point cloud from the estimated depth

This is highlighted in Figs. 3c and 3d. Finally, the Figs. 4b show the point clouds corresponding to Scene3 and Scene4 which are generated using the estimated depth values. This facilitates the usage of point processing libraries to refine the depth estimate by probabilistically fitting ground planes and building surfaces. Furthermore, such point processing algorithms can be incorporated in a closed loop fashion to enhance the structural consistency of the depth maps.

5 Discussion

In this paper we have presented a framework to estimate depths from two or more fisheye images. We employ variational technique to estimate the entire depth map at once. A novel surface area based term is used as a regularizer for the energy functional. As a result we get an anisotropic diffusion term which we believe facilitates in a more realistic surface reconstruction through the means of realistic depth estimates. The regularizer term also assists in establishing the Lax-Milgram property which is essential in proving existence of a unique weak solution in $H^1(\mathfrak{D})$. In order to ensure positivity of the solution and to facilitate the use of additional regularizing constraints we use primal-dual methods to solve (9) in least squares sense. The least-squares interpretation enables us to infer the 2nd order moment i.e. the covariance operator of the depth estimate. Since we assume Gaussian noise, covariance (along with the mean) is sufficient to estimate the entire distribution of the depth map. It should be noted that a least-squares interpretation is not a necessary step to have an uncertainty estimation, alternatively one can look at (9) as a stochastic pde which also enables one to estimate the uncertainty of the solution. We believe that uncertainty quantification is of paramount importance for many safety critical applications (e.g. robotics, autonomous driving). Complex applications requires interaction of various modules that work with different types of data. In such cases covariance of the estimate is a crucial information that facilitates in fusing heterogeneous data. Thereby, seamlessly integrating different higher level modules. Furthermore, an entire range of Bayesian algorithms become applicable for image based inferences. Other popular approaches (esp. deep learning based techniques) lack to provide any kind of inference about the distribution of the estimate, thus incorporation of these techniques into complex real-time systems puts the entire system at great risk. However, as mentioned above, such offline algorithms, that offer quick initial estimates, can be combined with our online algorithm to improve the overall reliability of the system.

References

1. Bettonvil, F.: Fisheye lenses. WGN J. Int. Meteor Organ. **33**(1), 9–14 (2005)
2. Caruso, D., Engel, J., Cremers, D.: Large-scale direct slam for omnidirectional cameras, pp. 141–148, September 2015. https://doi.org/10.1109/IROS.2015.7353366
3. Drulea, M., Szakats, I., Vatavu, A., Nedevschi, S.: Omnidirectional stereo vision using fisheye lenses, pp. 251–258, October 2014. https://doi.org/10.1109/ICCP.2014.6937005

4. Friel, M., Hughes, C., Denny, P., Jones, E., Glavin, M.: Automatic calibration of fish-eye cameras from automotive video sequences. Intell. Transp. Syst. IET **4**, 136–148 (2010). https://doi.org/10.1049/iet-its.2009.0052
5. Geiger, A., Roser, M., Urtasun, R.: Efficient large-scale stereo matching. In: Kimmel, R., Klette, R., Sugimoto, A. (eds.) ACCV 2010. LNCS, vol. 6492, pp. 25–38. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19315-6_3
6. Hirschmüller, H.: Stereo processing by semi-global matching and mutual information. IEEE Trans. Pattern Anal. Mach. Intell. **30**, 328–341 (2008)
7. Hughes, C., Glavin, M., Jones, E., Denny, P.: Wide-angle camera technology for automotive applications: a review. Intell. Transp. Syst. IET **3**, 19–31 (2009). https://doi.org/10.1049/iet-its:20080017
8. Hughes, C., O'Malley, R., O'Cualain, D., Glavin, M., Jones, E.: Trends towards automotive electronic vision systems for mitigation of accidents in safety critical situations, January 2011. https://doi.org/10.5772/12907
9. Häne, C., et al.: 3d visual perception for self-driving cars using a multi-camera system: calibration, mapping, localization, and obstacle detection. Image Vis. Comput. (2017). https://doi.org/10.1016/j.imavis.2017.07.003
10. Kakani, V., Kim, H., Kumbham, M., Park, D., Jin, C., Nguyen, V.: Feasible self-calibration of larger field-of-view (FOV) camera sensors for the advanced driver-assistance system (ADAS). Sensors **19**, 3369 (2019). https://doi.org/10.3390/s19153369
11. Keselman, L.: What's the best lens for stereo? An exploration of lens models and their stereoscopic depth performance. Stanford (2016). https://pdfs.semanticscholar.org/39cf/7c007e33725552d6242eef2f1c6713830812.pdf?_ga=2.238984768.1758024819.1591704773-1508129036.1591704773
12. Khomutenko, B., Garcia, G., Martinet, P.: An enhanced unified camera model. IEEE Robot. Autom. Lett. **1**, 1–1 (2015). https://doi.org/10.1109/LRA.2015.2502921
13. Kim, H., Hilton, A.: 3d scene reconstruction from multiple spherical stereo pairs. Int. J. Comput. Vis. **104** (2013). https://doi.org/10.1007/s11263-013-0616-1
14. Kosov, S., Thormählen, T., Seidel, H.-P.: Accurate real-time disparity estimation with variational methods. In: Bebis, G., et al. (eds.) ISVC 2009. LNCS, vol. 5875, pp. 796–807. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-10331-5_74
15. Kumar, V.R., et al.: FisheyeDistanceNet: self-supervised scale-aware distance estimation using monocular fisheye camera for autonomous driving (2019)
16. Li, S.: Binocular spherical stereo. IEEE Trans. Intell. Transp. Syst. **9**, 589–600 (2009). https://doi.org/10.1109/TITS.2008.2006736
17. Newcombe, R., Lovegrove, S., Davison, A.: DTAM: dense tracking and mapping in real-time, pp. 2320–2327, November 2011. https://doi.org/10.1109/ICCV.2011.6126513
18. Olson, L.N., Schroder, J.B.: PyAMG: algebraic multigrid solvers in Python v4.0 (2018). https://github.com/pyamg/pyamg. Release 4.0
19. Roxas, M., Oishi, T.: Variational fisheye stereo. IEEE Robot. Autom. Lett. **PP**, 1–1 (2020). https://doi.org/10.1109/LRA.2020.2967657
20. Schneider, J., Stachniss, C., Förstner, W.: On the accuracy of dense fisheye stereo. IEEE Robot. Autom. Lett. **1**(1), 227–234 (2016). https://doi.org/10.1109/LRA.2016.2516509
21. Schwalbe, E.: Geometric modelling and calibration of fisheye lens camera systems. In: International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. 36, May 2012

22. Shah, S., Aggarwal, J.K.: Depth estimation using stereo fish-eye lenses. In: Proceedings of 1st International Conference on Image Processing, vol. 2, pp. 740–744, November 1994. https://doi.org/10.1109/ICIP.1994.413669
23. Vieira, G., et al.: Stereo vision methods: from development to the evaluation of disparity maps, pp. 132–137, October 2017. https://doi.org/10.1109/WVC.2017.00030