Optimisation of classic photometric stereo by non-convex variational minimisation

Georg Radow, Laurent Hoeltgen, Yvain Quéau, Michael Breuss

manuscript No.
(will be inserted by the editor)

Optimisation of classic photometric stereo by non-convex variational minimisation

Georg Radow · Laurent Hoeltgen · Yvain Quéau · Michael Breuß

Received: date / Accepted: date

Abstract Estimating shape and appearance of a three dimensional object from a given set of images is a classic research topic that is still actively pursued. Among the various techniques available, photometric stereo is distinguished by the assumption that the underlying input images are taken from the same point of view but under different lighting conditions. The most common techniques are conceptually close to the classic photometric stereo problem, meaning that the modelling encompasses a linearisation step and that the shape information is computed in terms of surface normals. In this work, instead of linearising we aim to stick to the original formulation of the photometric stereo problem, and we propose to minimise a much more natural objective function, namely the reprojection error in terms of depth.

Minimising the resulting non-trivial variational model for photometric stereo allows to recover the depth of the photographed scene directly. As a solving strategy, we follow an approach based on a recently published optimisation scheme for non-convex and non-smooth cost functions.

The main contributions of our paper are of theoretical nature. A technical novelty in our framework is the usage of matrix differential calculus. We supplement our approach by a detailed convergence analysis of the resulting optimisation algorithm and discuss possibilities to ease the computational complexity. At hand of an experimental evaluation we discuss important properties of the method. Overall, our strategy achieves more accurate results than other approaches that rely on the classic photometric stereo assumptions. The experiments also highlight some practical aspects of the underlying optimisation algorithm that may be of interest in a more general context.

G. Radow · L. Hoeltgen · M. Breuß
Chair for Applied Mathematics, BTU Cottbus-Senftenberg, Cottbus, Germany.
email: {radow,hoeltgen,breuss}@b-tu.de

Y. Quéau
Vision Lab. ISEN Brest, L@bISEN, Brest, France.
email: yvain.queau@isen-ouest.yncrea.fr

1 Introduction

The reconstruction of three dimensional depth information given a set of two dimensional input images is a classic problem in computer vision. The class of methods fulfilling this task by inferring local shape from brightness analysis is called photometric methods [11,39]. They usually employ a static view point and variations in illumination to obtain the 3D structure. Fundamental photometric reconstruction processes are shape from shading (SFS) and photometric stereo (PS) [11]. Shape from shading typically requires a single input image, whereas PS makes use of several input images taken from a fixed view point under different illumination. Photometric stereo incorporates SFS in the sense that SFS equations applied to each of the input images are integrated into a common PS process in order to obtain the 3D shape. This integrated model is usually formulated as an optimisation task that best explains the input images in terms of a pointwise estimation of shape and appearance.

The pioneer of the PS method was Woodham in 1978 [40], see also Horn *et al.* [12]. The mathematical formulation of the PS problem is based on the use of the image irradiance equation (IIE) as in SFS for the individual input images, respectively. The image irradiance equation constitutes a relation between the image intensity and the reflectance map. The classic proceeding is thereby to consider Lambert's law [18] for modelling the appearance of a shape given information on its geometry and albedo as well as the lighting in a scene. It has been shown that the orientation of a Lambertian surface can be uniquely determined from the resulting appearance variations provided that the surface is illuminated by at least three known, non-coplanar light

sources, corresponding to at least three input images [41]. However, let us also mention the classic work of Kozera [17] as well as Onn and Bruckstein [26] where refined existence and uniqueness results are presented for the two-image case. As a beneficial aspect beyond the possible estimation of 3D shape, PS enables to compute an albedo map allowing to deal with non-uniform object materials or textured objects in a photographed scene.

As to complete our brief review of some general aspects of PS, let us note that it is possible to extend Woodham's classic PS model, for instance to non-Lambertian reflectance [2, 14, 16, 21, 38], or to take into account several types of lighting in a scene [3, 31]. One may also consider a PS approach based on solving partial differential equations (PDEs) corresponding to ratios of the underlying IIEs, see *e.g.* [22, 38]. The latter approach makes it possible to compute the 3D shape directly, whereas in most methods following the classic PS setting a field of surface normals is computed, which needs to be integrated in another step; see *e.g.* [1] for a recent discussion of integration techniques.

Let us turn to the formulation of the PS approach we make use of. At this stage we keep the presentation rather general as we elaborate afterwards in Section 2 on the details that are of some importance in the context of applying our optimisation approach.

Photometric 3D reconstruction is often formulated as an inverse problem: given an image I, the aim is to compute a depth map z that best explains the observed grey levels of the data. To this end we use the IIE $I(u,v) = \mathscr{R}(\mathbf{n}(u,v); \mathbf{s}, \rho(u,v))$ where $(u,v) \in \Omega$ represent the coordinates over the reconstruction domain $\Omega \subseteq \mathbb{R}^2$, $\mathbf{n}(u,v)$ denotes normal vectors to the surface z and $\mathscr{R}$ denotes the reflectance map [11]. This model describes interactions between the surface z and the lighting $\mathbf{s}$. The vector ρ represents reflectance parameters as *e.g.* the albedo, which can be either known or considered as hidden unknown parameters. For the sake of simplicity, we will consider in this paper only Lambertian reflectance without shadows, and we assume that the lighting of a photographed scene is directional and known. Moreover our camera is assumed to perform an orthographic projection. As in PS several input images I^i, $i \in \{1,\dots,m\}$ are considered under varying lightings $\mathbf{s}^i$, $i \in \{1,\dots,m\}$, the PS problem consists in finding a depth map z in terms of its surface normals $\mathbf{n}$ that best explains all IIEs simultaneously:

$$I^i(u,v) = \mathscr{R}(\mathbf{n}(u,v); \mathbf{s}^i, \rho)\,, \qquad i \in \{1,\dots,m\}\,. \tag{1}$$

Our contribution. Existing methods are based on linearising the PS model, either through the estimation of scaled normals followed by integration, or by differential ratios. However the optimal solution to the linearised model is different from that of the original model, which is not linear. Our aim is to stick to the original model as close as possible. Thus we strive to obtain the solution z of the PS problem directly, see also Figure 1 for an account. We show that estimating the optimal solution necessarily involves non-trivial optimisation methods, even with the simplest models for the reflectance function $\mathscr{R}$ and the most simple deviations from the model assumptions that may occur, *i.e.* we consider Lambertian reflectance without shadows and additive, zero-mean Gaussian noise.

To achieve our goal we propose a numerical framework to approximate an optimal solution which can be used to refine classic PS results. Our approach relies on matrix differential theory for analytic derivations and on recent developments in non-convex optimisation. In that novel framework for this class of problems we prove here the convergence of the optimisation method. The theoretical results are supplemented by a thorough numerical investigation that highlights some important observations on the optimisation routine.

The basic procedure of this work has been the subject of our conference paper [10], the results of which are mainly contained in the second, third and beginning of the fourth section of this article. Our current paper extends that previous work significantly by providing the mathematical validation of convergence and the extended analysis of the numerical optimisation algorithm. These are also exactly the core contributions of this paper. Moreover, we give a much more detailed description of the matrix calculus framework we employ and provide additional experiments.

2 Construction of our method and more related work

As shown by Woodham [41], all surface normals can be estimated in the classic PS model without ambiguity, provided $m \geq 3$ input images and non-coplanar calibrated lighting vectors are given. In addition, the reflectance parameters (*e.g.* the albedo) can also be estimated. This is achieved usually by minimising the difference between the given data, *i.e.* the input images and the reprojection according to the estimated normal and albedo:

$$\arg\min_{\mathbf{n},\rho} \frac{1}{m} \iint_\Omega \sum_{i=1}^{m} \Phi\left(I^i - \mathscr{R}\left(\mathbf{n}; \mathbf{s}^i, \rho\right)\right) \mathrm{d}u\,\mathrm{d}v\,, \tag{2}$$

with a penaliser Φ. As a result, one obtains an approximation of the normal $\mathbf{n}(u,v)$ and the albedo $\rho(u,v)$ at each position (u,v).

Since there is no coupling between the normals estimated in two neighboring pixels, those estimates are the optimal *local* explanations of the image, in the sense of the estimator Φ. Yet, the estimated normal field is in general not integrable. Thus, the depth map that can be obtained by integration is not an optimal image explanation, but only a smooth explanation of the noisy normal field, *c.f.* Figure 1.

Example input image for PS | Classic PS with integration | Our method

Fig. 1 From a set of $m \geq 3$ images (*c.f.* left), classic PS provides an albedo and a normal map which best explain the input images in the sense of a local, pointwise estimation. In a second step, the smooth depth map is estimated by integration. Yet, the final surface is not the best explanation of the images, as indicated by the reprojection error (*c.f.* energy in (5)) (middle). We display this using white for $2.5 \cdot 10^{-3}$ and black for zero. Instead of this local procedure, we propose to minimise the reprojection criterion in terms of the depth and the albedo, through global non-convex optimisation. Not only the images are better explained (right), but we also demonstrate that the 3D-reconstruction results are improved (*c.f.* Section 6).

Instead of this pointwise joint estimation of the normal and the albedo, it is, as already mentioned in the introduction, possible to employ photometric ratios. Following that procedure means to divide the i-th by the j-th IIE in (1). This way, one obtains a homogeneous linear system in each normal vector that does not depend on the albedo, see [22]. However, these ratios introduce additional difficulties in the models. It is common to assume that image data is corrupted by additive, zero-mean, Gaussian noise. In that case the maximum likelihood (ML) function is consequently given by a quadratic function. Unfortunately, the ratio of two Gaussian random variables follows a Cauchy distribution [9]. Thus, additional care has to be taken when considering such distribution. Another frequent assumption is that the estimated normal fields should be integrable, yet, this is a rather restrictive assumption. The normal field computed by many aforementioned PS approaches does not necessarily need to be integrable. Hence, the integration task is usually formulated as another optimisation problem which aims at minimising the discrepancy between the estimated normal field and that of the recovered surface. Following that approach we now go into some more details.

Assuming orthographic camera projection, the relation between the normal $\mathbf{n}(u,v)$ and the depth $z(u,v)$ is given by:

$$\mathbf{n}(u,v) \coloneqq \frac{1}{\sqrt{\|\nabla z(u,v)\|^2+1}} \left[-\nabla z(u,v),\, 1\right]^\top , \tag{3}$$

where ∇z is the gradient of z. Then, the best smooth surface explaining the computed normals can be estimated in several ways [1], for instance by solving the variational problem:

$$\arg\min_{z} \iint_\Omega \Psi\left(\left\| \nabla z + \begin{bmatrix} \mathbf{n}_1/\mathbf{n}_3 \\ \mathbf{n}_2/\mathbf{n}_3 \end{bmatrix} \right\|^2 \right) \mathrm{d}u\,\mathrm{d}v \,, \tag{4}$$

where Ψ is again some estimator function; see [6, 8] for some discussion.

One may realise that, at this stage of the process chain of PS with integration, the images are not explicitly considered anymore. Thus, the final surface is in general not necessarily optimal in the sense of the reprojection criterion. Regularising the normal field before integration [35,43] may also ensure integrability, but since such methods only use the normal field, and not the images, they may be unable to assert optimality with respect to the reprojection.

Global PS approaches solve the latter problem as they represent a way to ensure that the recovered surface is optimal with respect to the reprojection criterion. Moreover, it is possible to solve the system (1) directly in terms of the depth [5]: this ensures both that the recovered surface is regular, and that it is optimal with respect to the reprojection criterion, calculated from the depth map z and not from a non-integrable estimate of its gradient. Some PDE-based PS approaches have been recently proposed, and were shown to ease the resolution in particularly difficult situations such as pointwise lighting [31] and specular reflectance [38]. To ensure robustness, such methods can be coupled with variational methods. In other words, the criterion which should be considered for ensuring optimality of a surface reconstruction by PS is not the local criterion (2), but rather

$$\arg\min_{z,\rho} \frac{1}{m} \iint_\Omega \sum_{i=1}^{m} \Phi\left(I^i - \mathscr{R}\left(z; \mathbf{s}^i, \rho\right)\right) \mathrm{d}u\,\mathrm{d}v \,, \tag{5}$$

where $\mathscr{R}$ is now a function of the depth map z through the relation (3). A theoretical analysis of the choice $\Phi(x) = |x|$, can be found in [4]. Numerical resolution methods based on proximal splittings were more recently introduced in [32].

Yet, this last work relies on an "optimise then discretise" approach which would involve non-trivial oblique boundary conditions (BC), replaced there for simplicity reasons by Dirichlet BC. Obviously, this represents a strong limitation which prevents working with many real-world data where this oblique BC is rarely available.

The optimisation problem (5) is usually non-linear and non-convex. The ratio procedure described earlier can be used: it simultaneously eliminates the albedo and the non-linear terms, *c.f.* [7, 21, 37, 38] and obviously removes the bias due to non-integrability. But let us recall that it is only the best linear unbiased estimate, and also not the optimal one. To guarantee optimality, it is necessary to minimise the nonquadratic, non-convex energy, *i.e.* without employing ratios. Other methods [31, 34] overcome the nonlinearity by absorbing it in the auxiliary albedo variable. Again, the solution is not that of the original problem (5) which remains, to the best of our knowledge, unsolved.

Solving (5) is a challenging problem. Efficient strategies to find the sought minimum are scarce. Recently Ochs *et al.* [25] proposed a novel method to handle such non-convex optimisation problems, called *iPiano*. A major asset of the approach is the extensive convergence theory provided in [24, 25]. Because of this solid mathematical foundation we explore the iPiano approach in this work. The scheme makes explicit use of the derivative of the cost function, which in our case involves derivatives of matrix-valued functions, and we will employ as a technical novelty, matrix differential theory [19, 20] to derive the resulting scheme.

3 Non-convex discrete variational model for PS

In this section we describe the details of our framework for estimating both the depth and the (Lambertian) reflectance parameters over the domain Ω.

3.1 Assumptions on the PS model

We assume $m \geq 3$ grey level images I^i, $i \in \{1,\dots,m\}$, are available, along with the m lighting vectors $\mathbf{s}^i \in \mathbb{R}^3$, assumed to be known and non-coplanar. Let us first go back a step to using surface normals. We assume Lambertian reflectance and neglect shadows, which leads to the following well-known model:

$$\mathscr{R}\left(\mathbf{n}(u,v);\mathbf{s}^i,\rho\right) \coloneqq \rho\,(u,v)\left\langle \mathbf{s}^i, \mathbf{n}\,(u,v)\right\rangle, \tag{6}$$

where $(u,v) \in \Omega$, $i = 1,\dots,m$ and $\rho(u,v)$ is the albedo at the surface point conjugated to position (u,v), considered as a hidden unknown parameter. Let us note that real-world PS images can be processed by low-rank factorisation techniques in order to match the linear reflectance model (6), *c.f.* [42].

We further assume orthographic projection, hence the normal $\mathbf{n}(u,v)$ is given by (3). Then the reflectance model becomes a function of the depth map z:

$$\mathscr{R}\left(z;\mathbf{s}^i,\rho\right) \coloneqq \frac{\rho\,(u,v)}{\sqrt{\|\nabla z(u,v)\|^2+1}}\left\langle \mathbf{s}^i, \begin{bmatrix} -\nabla z(u,v) \\ 1 \end{bmatrix}\right\rangle, \tag{7}$$

with $(u,v) \in \Omega$, for all i. Eventually, we assume that the images I^i differ from this reflectance model only up to additive, zero-mean, Gaussian noise. The ML estimator is thus the least-squares estimator $\Phi(x) = \frac{1}{2}x^2$, and the cost function in the reprojection criterion (5) becomes:

$$\mathscr{E}_{\mathscr{R}}\left(z,\rho;I\right) \coloneqq \frac{1}{2m}\iint_{\Omega}\sum_{i=1}^{m}\left(I^i - \mathscr{R}\left(z;\mathbf{s}^i,\rho\right)\right)^2 \mathrm{d}u\,\mathrm{d}v\,. \tag{8}$$

3.2 Tikhonov regularisation of the model

Our energy in (8) only depends on the gradient ∇z and not on the depth $z(u,v)$ itself. As a consequence, solutions can only be determined up to an arbitrary constant. As a remedy we follow [21] and introduce a reference depth $z_0(u,v)$, thus regularising our initial model with a zero-th order Tikhonov regulariser controlled by a parameter $\lambda > 0$:

$$\underset{z,\rho}{\arg\min}\,\mathscr{E}_{\mathscr{R}}\left(z,\rho;I\right) + \frac{\lambda}{2}\iint_{\Omega}\left(z-z_0\right)^2 \mathrm{d}u\,\mathrm{d}v\,. \tag{9}$$

In practice, λ can be set to any small value, so that a solution of (9) lies as close as possible to a minimiser of (8). In our experiments we set $\lambda \coloneqq 10^{-6}$, if not specified otherwise, and z_0 as the classic PS solution followed by least-squares integration [1]. We remark that the numerical condition of the problem depends on λ, and thus this parameter influences the convergence rate of numerical schemes. An experimental study of λ is part of Section 6.2.

3.3 Discretisation

As already mentioned, "optimise then discretise" approaches for solving (9), such as [32], involve non-trivial BC. Hence, we prefer a "discretise then optimise", finite dimensional formulation of the variational PS problem (9).

In our discrete setting we are given m images $\mathbf{I}^i$, $i \in \{1,\dots,m\}$, with n pixels labelled with a single index j running from 1 to n. We discretise (9) in the following way:

$$\underset{\mathbf{z},\rho\in\mathbb{R}^n}{\arg\min}\left\{\frac{1}{2m}\sum_j\left\|\mathbf{I}_j - \frac{\rho_j}{\sqrt{\left\|\nabla\mathbf{z}_j\right\|^2+1}}\mathbf{S}\begin{bmatrix}-\nabla\mathbf{z}_j\\1\end{bmatrix}\right\|^2 + \frac{\lambda}{2}\left\|\mathbf{z}_j-\mathbf{z}_{0j}\right\|^2\right\}. \tag{10}$$

where $\mathbf{I}_j \coloneqq [\mathbf{I}_j^1, \dots, \mathbf{I}_j^m]^\top \in \mathbb{R}^m$ is the vector of intensities at pixel j, $\nabla \mathbf{z}_j$ represents now a finite difference approximation of the gradient of $\mathbf{z}$ at pixel j, and $\mathbf{S} = [\mathbf{s}^1, \dots, \mathbf{s}^m]^\top \in \mathbb{R}^{m,3}$ is a matrix containing the stacked m lighting vectors $\mathbf{s}^i$.

We remark that the matrix $\mathbf{S}$ can be decomposed into two sub-matrices $\mathbf{S}_\ell$ and $\mathbf{S}_r$ of dimensions $m \times 2$ and $m \times 1$ such that $\mathbf{S} \coloneqq \begin{bmatrix}\mathbf{S}_\ell & \mathbf{S}_r\end{bmatrix}$, and so that

$$\mathbf{S} \begin{bmatrix} -\nabla \mathbf{z}_j \\ 1 \end{bmatrix} = -\mathbf{S}_\ell \nabla \mathbf{z}_j + \mathbf{S}_r \,. \tag{11}$$

Let us also introduce a $2n \times n$ block matrix $\mathbf{M}$, such that each block $\mathbf{M}_j$ is a $2 \times n$ matrix containing the finite difference coefficients used for approximating the gradient:

$$\mathbf{M} \coloneqq \begin{bmatrix}\mathbf{M}_1 & \dots & \mathbf{M}_n\end{bmatrix}^\top \in \mathbb{R}^{2n,n} \,, \qquad \mathbf{M}_j \mathbf{z} = \nabla \mathbf{z}_j \in \mathbb{R}^2 \,. \tag{12}$$

We further introduce the aliases

$$\mathbf{A}_j(\mathbf{z}, \rho) \coloneqq -\frac{\rho_j}{\sqrt{1 + \left\|\mathbf{M}_j \mathbf{z}\right\|^2}} \mathbf{S}_\ell \in \mathbb{R}^{m,2} \tag{13}$$

and

$$\mathbf{b}_j(\mathbf{z}, \rho) \coloneqq \mathbf{I}_j - \frac{\rho_j}{\sqrt{1 + \left\|\mathbf{M}_j \mathbf{z}\right\|^2}} \mathbf{S}_r \in \mathbb{R}^m \,, \tag{14}$$

and stack them, respectively, in a block-diagonal matrix

$$\mathbf{A}(\mathbf{z}, \rho) \coloneqq \begin{bmatrix} \mathbf{A}_1(\mathbf{z}, \rho) & & \\ & \ddots & \\ & & \mathbf{A}_n(\mathbf{z}, \rho) \end{bmatrix} \in \mathbb{R}^{mn,2n} \tag{15}$$

and a vector

$$\mathbf{b}(\mathbf{z}, \rho) \coloneqq \begin{bmatrix} \mathbf{b}_1(\mathbf{z}, \rho) \\ \vdots \\ \mathbf{b}_n(\mathbf{z}, \rho) \end{bmatrix} \in \mathbb{R}^{mn} \,. \tag{16}$$

Using these notational conventions as well as

$$f(\mathbf{z}, \rho) \coloneqq \frac{1}{2m} \|\mathbf{A}(\mathbf{z}, \rho)\mathbf{M}\mathbf{z} - \mathbf{b}(\mathbf{z}, \rho)\|_2^2 \tag{17}$$

and

$$g(\mathbf{z}) \coloneqq \frac{\lambda}{2} \|\mathbf{z} - \mathbf{z}_0\|_2^2 \tag{18}$$

the task in (10) can be rewritten compactly as

$$\operatorname*{arg\,min}_{\mathbf{z}, \rho \in \mathbb{R}^n} \{f(\mathbf{z}, \rho) + g(\mathbf{z})\} \,. \tag{19}$$

which is the discrete PS model we propose to tackle in this paper. Observe that, if $\mathbf{A}(\mathbf{z}, \rho)$ and $\mathbf{b}(\mathbf{z}, \rho)$ were constant, problem (19) would be a linear least squares problem with respect to $\mathbf{z}$.

Let us remark that (19) can be easily extended to include more realistic reflectance [15, 16] and lighting [28, 31] models, as well as more robust estimators [14, 34]: this only requires to change the definition of f, which stands for the global reprojection error $\mathscr{E}_\mathscr{R}$. However, the adaptation of the optimisation, which for the model (19) is discussed in Sections 4 and 5, may not be straightforward.

3.4 Alternating optimisation strategy

In order to ensure applicability of our method to real-world data, the albedo ρ cannot be assumed to be known. Inspired by the well-known Expectation-Maximisation algorithm, we treat ρ as a hidden parameter, and opt for an alternating strategy which iteratively refines the depth with fixed albedo, and the hidden parameter with fixed depth:

$$\mathbf{z}^{(k+1)} = \operatorname*{arg\,min}_{\mathbf{z}} \left\{ f\left(\mathbf{z}, \rho^{(k)}\right) + g(\mathbf{z}) \right\} , \tag{20}$$

$$\rho^{(k+1)} = \operatorname*{arg\,min}_{\rho} \left\{ f\left(\mathbf{z}^{(k+1)}, \rho\right) + g\left(\mathbf{z}^{(k+1)}\right) \right\} , \tag{21}$$

starting from $\mathbf{z}^{(0)} = \mathbf{z}_0$ and taking as $\rho^{(0)}$ the albedo obtained by the classic PS approach [41]. Of course, the choice of a particular prior $\mathbf{z}_0$ has a direct influence on the convergence of the algorithm. The proposed scheme is guaranteed to converge, even with a trivial prior $\mathbf{z}_0 \equiv$ constant. The alternating optimisation creates a decreasing sequence of energy values which is bounded below by zero. Thus there exists a converging subsequence. However, since f and thereby also the energy in (19) are non-convex, we can only expect convergence towards a local minimiser. Thus the proposed method should be considered as a post-processing technique to refine classic PS approaches, rather than as a standalone PS method.

Now, let us comment on the two optimisation problems in (20). Updating ρ amounts pointwise to a linear least-squares problem, which admits the following closed-form solution at each pixel:

$$\rho_j^{(k+1)} = \frac{\sqrt{1 + \|\mathbf{M}_j \mathbf{z}^{(k+1)}\|^2} \sum_{i=1}^{m} \mathbf{I}_j^i \mathbf{s}^{i\top} \begin{bmatrix} -\mathbf{M}_j \mathbf{z}^{(k+1)} \\ 1 \end{bmatrix}}{\sum_{i=1}^{m} \left(\mathbf{s}^{i\top} \begin{bmatrix} -\mathbf{M}_j \mathbf{z}^{(k+1)} \\ 1 \end{bmatrix} \right)^2} \,. \tag{22}$$

The computation of $\mathbf{z}^{(k+1)}$ is considerably harder, and it is dealt with in the following paragraphs.

4 An inertial proximal point algorithm for PS

In this section we discuss the numerical solution strategy of our problem (20). We especially discuss the main difficulty within this strategy, that is to compute the gradient with respect to $\mathbf{z}$ for the function f in (17). Apart from the explicit formula for this gradient we also investigate a CPU based approximation leading to efficient computations on a desktop computer.

4.1 The iPiano algorithm

We will now make precise the iPiano algorithm [25] for our problem (20). Since the albedo is fixed for the purpose

of the corresponding optimisation stage, we denote $f(\mathbf{z}) = f(\mathbf{z}, \boldsymbol{\rho}^{(k)})$. The iPiano algorithm seeks a minimiser of

$$\min_{\mathbf{x}\in\mathbb{R}^n} \{f(\mathbf{x}) + g(\mathbf{x})\} , \tag{23}$$

where $g\colon \mathbb{R}^n \to \mathbb{R}$ is convex and $f\colon \mathbb{R}^n \to \mathbb{R}$ is smooth. What makes iPiano appealing is the fact that g must not necessarily be smooth and f is not required to be convex. This allows manifold designs of novel fixed-point schemes. In its general form it evaluates

$$\operatorname{prox}_{\alpha g}\left(\mathbf{z}^{(k)} - \alpha\nabla f(\mathbf{z}^{(k)}) + \beta(\mathbf{z}^{(k)} - \mathbf{z}^{(k-1)})\right) , \tag{24}$$

where the proximal operator is given by

$$\operatorname{prox}_{\alpha g}(\mathbf{z}) \coloneqq \arg\min_{\mathbf{x}} \left\{ \frac{1}{2}\|\mathbf{x}-\mathbf{z}\|^2 + \alpha g(\mathbf{x}) \right\} \tag{25}$$

and goes back to Moreau [23]. Before we can define the final algorithm we also need to determine the gradient of f.

4.2 Matrix Calculus

First we recall some general rules to derive the Jacobian of a matrix valued function, before we apply these rules to our setting in the next section.

In our setting the main difficulty is that the matrix $\mathbf{A}$ depends on our sought unknown $\mathbf{z}$. In order to state a useful representation of arising differential expressions we have to resort to matrix differential calculus. We refer to [19, 20, 29, 30] for a more in-depth representation. A key notion is the definition of the Jacobian of a matrix, which can be obtained in several ways. In this paper we follow the one given in [19].

Definition 1 (Jacobian of a Matrix Valued Function) Let $\mathbf{A}$ be a differentiable $m \times p$ real matrix valued function of an $n \times q$ matrix $\mathbf{X}$ of real variables, *i.e.* $\mathbf{A} = \mathbf{A}(\mathbf{X})$. The *Jacobian matrix* of $\mathbf{A}$ at $\mathbf{X}$ is the $mp \times nq$ matrix

$$D[\mathbf{A}](\mathbf{X}) \coloneqq \frac{\mathrm{d}\operatorname{vec}(\mathbf{A}(\mathbf{X}))}{\mathrm{d}(\operatorname{vec}\mathbf{X})^\top} , \tag{26}$$

where $\operatorname{vec}(\cdot)$ corresponds to the vectorisation operator described in [13] (Definition 4.29). This operator stacks columnwise all the entries from its matrix argument to form a large vector.

Here, differentiability of a matrix valued function means that the corresponding vectorised function is differentiable in the usual sense. By this definition the computation of a matrix Jacobian can be reduced to computing a Jacobian for a vector valued function.

Example 1 Let $\mathbf{A}(\mathbf{x}) \in \mathbb{R}^{m,m}$ be a differentiable matrix valued function in diagonal form

$$\mathbf{A}(\mathbf{x}) = \begin{bmatrix} a_1(\mathbf{x}) & & \\ & \ddots & \\ & & a_m(\mathbf{x}) \end{bmatrix} \qquad \text{for all } \mathbf{x} \in \mathbb{R}^n , \tag{27}$$

then the Jacobian matrix of $\mathbf{A}$ at $\mathbf{x}$ has the form

$$D[\mathbf{A}](\mathbf{x}) = \frac{\mathrm{d}}{\mathrm{d}\mathbf{x}^\top}\begin{bmatrix} a_1(\mathbf{x}) \\ \mathbf{0}_{m,1} \\ a_2(\mathbf{x}) \\ \mathbf{0}_{m,1} \\ \vdots \\ a_m(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} \frac{\partial a_1(\mathbf{x})}{\partial \mathbf{x}_1} & \cdots & \frac{\partial a_1(\mathbf{x})}{\partial \mathbf{x}_n} \\ \mathbf{0}_{m,n} & & \\ \frac{\partial a_2(\mathbf{x})}{\partial \mathbf{x}_1} & \cdots & \frac{\partial a_2(\mathbf{x})}{\partial \mathbf{x}_n} \\ \mathbf{0}_{m,n} & & \\ \vdots & & \\ \frac{\partial a_m(\mathbf{x})}{\partial \mathbf{x}_1} & \cdots & \frac{\partial a_m(\mathbf{x})}{\partial \mathbf{x}_n} \end{bmatrix} , \tag{28}$$

where $\mathbf{0}_{p,q}$ denotes a $p \times q$ block of zeros.

The following two lemmas state extensions of the product and chain-rule to matrix valued settings. They provide us closed form representations that will be useful for the forthcoming findings. These results have been extracted from [19] (Theorem 7 and 9 respectively). Since these lemmas have been copied verbatim, we refer to their source for the detailed proofs.

Lemma 1 (Chain Rule) *Let S be a subset of $\mathbb{R}^{n,q}$ and assume that $\mathbf{F}\colon S \to \mathbb{R}^{m,p}$ is differentiable at an interior point $\mathbf{C}$ of S. Let T be a subset of $\mathbb{R}^{m,p}$ such that $\mathbf{F}(\mathbf{X}) \in T$ for all $\mathbf{X} \in S$, and assume that $\mathbf{G}\colon T \to \mathbb{R}^{r,s}$ is differentiable at an interior point $\mathbf{B} = \mathbf{F}(\mathbf{C})$ of T. Then the composite function $\mathbf{H}\colon S \to \mathbb{R}^{r,s}$ defined by $\mathbf{H}(\mathbf{X}) = \mathbf{G}(\mathbf{F}(\mathbf{X}))$ is differentiable at $\mathbf{C}$ and*

$$D[\mathbf{H}](\mathbf{C}) = D[\mathbf{G}](\mathbf{B})D[\mathbf{F}](\mathbf{C}) . \tag{29}$$

Definition 2 (Kronecker Product) Let $\mathbf{A} = (a_{i,j})$ be a $m \times n$ matrix and $\mathbf{B}$ be a $p \times q$ matrix then the Kronecker Product $\mathbf{A} \otimes \mathbf{B}$ is defined as

$$\mathbf{A} \otimes \mathbf{B} \coloneqq \begin{bmatrix} a_{1,1}\mathbf{B} & \cdots & a_{1,n}\mathbf{B} \\ \vdots & & \vdots \\ a_{m,1}\mathbf{B} & \cdots & a_{m,n}\mathbf{B} \end{bmatrix} . \tag{30}$$

Example 2 For a row vector $\mathbf{B} = [b_1, \ldots, b_n] \in \mathbb{R}^{1,n}$ and the identity matrix $\mathbf{1}_3 \in \mathbb{R}^{3,3}$ we have

$$\mathbf{B} \otimes \mathbf{1}_3 = \begin{bmatrix} b_1 & b_2 & & b_n & \\ & b_1 & b_2 & \cdots & b_n \\ & & b_1 & b_2 & b_n \end{bmatrix} . \tag{31}$$

Lemma 2 (Product Rule) *Let $\mathbf{U}\colon S \to \mathbb{R}^{m,r}$ and $\mathbf{V}\colon S \to \mathbb{R}^{r,p}$ be two matrix valued functions defined and differentiable on an open set $S \subseteq \mathbb{R}^{n,q}$. Then the matrix product $\mathbf{UV}\colon S \to \mathbb{R}^{m,p}$ is differentiable on S and the Jacobian matrix $D[\mathbf{UV}](\mathbf{X}) \in \mathbb{R}^{mp,nq}$ is given by*

$$D[\mathbf{UV}](\mathbf{X}) = (\mathbf{V}^\top \otimes \mathbf{1}_m)D[\mathbf{U}](\mathbf{X}) + (\mathbf{1}_p \otimes \mathbf{U})D[\mathbf{V}](\mathbf{X}) . \tag{32}$$

Here, $\mathbf{1}_k$ represents the identity matrix in $\mathbb{R}^{k,k}$.

Example 3 Let $\mathbf{A}$ be a differentiable $m \times m$-matrix valued function and $\mathbf{M}$ be a $m \times n$-matrix, then by Lemma 2 we have

$$\begin{aligned} &D\left[\mathbf{AMx}\right](\mathbf{x}) \\ &= \left((\mathbf{Mx})^\top \otimes \mathbf{1}_m\right) D\left[\mathbf{A}\right](\mathbf{x}) + (\mathbf{1}_1 \otimes \mathbf{A}(\mathbf{x}))\, D\left[\mathbf{Mx}\right](\mathbf{x}) \\ &= \left((\mathbf{Mx})^\top \otimes \mathbf{1}_m\right) D\left[\mathbf{A}\right](\mathbf{x}) + \mathbf{A}(\mathbf{x})\mathbf{M}\,. \end{aligned} \tag{33}$$

4.3 Gradient computation

The following two corollaries are a direct consequence from the foregoing statements. It suffices to plug in the corresponding quantities. We also remind, that our choice of the matrix derivative allows us to interpret vectors as matrices having a single column only.

Corollary 1 *Let $\mathbf{A}(\mathbf{z})$ be a $n \times q$ matrix valued function depending on $\mathbf{z} \in \mathbb{R}^m$ and $\mathbf{M} \in \mathbb{R}^{q,m}$ a matrix which does not depend on $\mathbf{z}$, then the Jacobian of the matrix-vector product $\mathbf{A}(\mathbf{z})\mathbf{Mz}$ is given by*

$$D[\mathbf{AMz}](\mathbf{z}) = \left((\mathbf{Mz})^\top \otimes \mathbf{1}_n\right) D[\mathbf{A}](\mathbf{z}) + \mathbf{A}(\mathbf{z})\mathbf{M}\,. \tag{34}$$

Proof We apply the product rule on the product between $\mathbf{A}(\mathbf{z})$ and $\mathbf{Mz}$ and subsequently on the product $\mathbf{Mz}$. In a first step this yields

$$D[\mathbf{AMz}] = \left((\mathbf{Mz})^\top \otimes \mathbf{1}_n\right) D[\mathbf{A}](\mathbf{z}) + \mathbf{A}(\mathbf{z}) D[\mathbf{Mz}](\mathbf{z})\,. \tag{35}$$

Since $D[\mathbf{Mz}](\mathbf{z}) = \mathbf{M}$ the result follows immediately. □

Corollary 2 and Theorem 1 yield our desired compact representations that we use for the algorithmic presentation of our iterative schemes.

Corollary 2 *Using the same assumptions as in Corollary 1, we deduce from the chain rule given in Lemma 1 the following relationship*

$$\begin{aligned} &\nabla \|\mathbf{A}(\mathbf{z})\mathbf{Mz}\|_2^2 \\ &= 2\underbrace{\left(D[\mathbf{A}](\mathbf{z})^\top \left((\mathbf{Mz}) \otimes \mathbf{1}_n\right) + \mathbf{M}^\top \mathbf{A}(\mathbf{z})^\top\right)}_{= D[\mathbf{AMz}](\mathbf{z})^\top} \mathbf{A}(\mathbf{z})\mathbf{Mz}\,, \end{aligned} \tag{36}$$

where ∇ denotes the gradient with respect to $\mathbf{z}$.

Proof Since $D\left[\|\mathbf{x}\|_2^2\right](\mathbf{x})$ is given by $2\mathbf{x}^\top$ we conclude from the chain- and product-rule that

$$\begin{aligned} &D[\|\mathbf{AMz}\|_2^2](\mathbf{z}) \\ &= 2\big(\mathbf{A}(\mathbf{z})\mathbf{Mz}\big)^\top D[\mathbf{AMz}](\mathbf{z}) \\ &= 2\big(\mathbf{A}(\mathbf{z})\mathbf{Mz}\big)^\top \\ &\quad \Big(\big((\mathbf{Mz})^\top \otimes \mathbf{1}_n\big) D[\mathbf{A}](\mathbf{z}) + \mathbf{A}(\mathbf{z}) D[\mathbf{Mz}](\mathbf{z})\Big)\,. \end{aligned} \tag{37}$$

Since the gradient is simply the transposed version of the Jacobian, we obtain

$$\begin{aligned} \nabla \|\mathbf{A}(\mathbf{z})\mathbf{Mz}\|_2^2 = 2\Big(&\Big((\mathbf{Mz})^\top \otimes \mathbf{1}_n\Big) D[\mathbf{A}](\mathbf{z}) \\ &+ \mathbf{A}(\mathbf{z}) D[\mathbf{Mz}](\mathbf{z})\Big)^\top \mathbf{A}(\mathbf{z})\mathbf{Mz}\,, \end{aligned} \tag{38}$$

from which the statement follows immediately. □

Let us now come to our main result.

Theorem 1 *Let $f(\mathbf{z}) = \|\mathbf{A}(\mathbf{z})\mathbf{Mz} - \mathbf{b}(\mathbf{z})\|_2^2$ be given with continuously differentiable data $\mathbf{A}(\mathbf{z})$ and $\mathbf{b}(\mathbf{z})$. Then we have for the gradient of f the following closed form expression:*

$$\begin{aligned} \nabla f(\mathbf{z}) = 2\Big(&\mathbf{A}(\mathbf{z})\mathbf{M} + \Big((\mathbf{Mz})^\top \otimes \mathbf{1}_n\Big) D[\mathbf{A}](\mathbf{z}) - D[\mathbf{b}](\mathbf{z})\Big)^\top \\ &(\mathbf{A}(\mathbf{z})\mathbf{Mz} - \mathbf{b}(\mathbf{z}))\,. \end{aligned} \tag{39}$$

Proof From the relationship between the canonical scalar product in $\mathbb{R}^n$ and the Euclidean norm we deduce that

$$\begin{aligned} &\|\mathbf{A}(\mathbf{z})\mathbf{Mz} - \mathbf{b}(\mathbf{z})\|_2^2 \\ &\quad = \|\mathbf{A}(\mathbf{z})\mathbf{Mz}\|_2^2 + \|\mathbf{b}(\mathbf{z})\|_2^2 - 2\langle \mathbf{A}(\mathbf{z})\mathbf{Mz}, \mathbf{b}(\mathbf{z})\rangle\,. \end{aligned} \tag{40}$$

Applying the gradient at each term separately and using the results from the previous corollaries, we obtain

$$\begin{aligned} &\nabla \|\mathbf{A}(\mathbf{z})\mathbf{Mz} - \mathbf{b}(\mathbf{z})\|_2^2 \\ &= 2\underbrace{\left(D[\mathbf{A}](\mathbf{z})^\top \left((\mathbf{Mz}) \otimes \mathbf{1}_n\right) + \mathbf{M}^\top \mathbf{A}(\mathbf{z})^\top\right)}_{= D[\mathbf{AMz}](\mathbf{z})^\top} \mathbf{A}(\mathbf{z})\mathbf{Mz} \\ &\quad + 2D[\mathbf{b}](\mathbf{z})^\top \mathbf{b}(\mathbf{z}) \\ &\quad - 2\underbrace{\left(D[\mathbf{A}](\mathbf{z})^\top \left((\mathbf{Mz}) \otimes \mathbf{1}_n\right) + \mathbf{M}^\top \mathbf{A}(\mathbf{z})^\top\right)}_{= D[\mathbf{AMz}](\mathbf{z})^\top} \mathbf{b}(\mathbf{z}) \\ &\quad - 2D[\mathbf{b}](\mathbf{z})^\top \mathbf{A}(\mathbf{z})\mathbf{Mz}\,, \end{aligned} \tag{41}$$

which can be simplified to

$$\begin{aligned} &\nabla \|\mathbf{A}(\mathbf{z})\mathbf{Mz} - \mathbf{b}(\mathbf{z})\|_2^2 \\ &\quad = 2\big(D[\mathbf{AMz}](\mathbf{z}) - D[\mathbf{b}](\mathbf{z})\big)^\top \big(\mathbf{A}(\mathbf{z})\mathbf{Mz} - \mathbf{b}(\mathbf{z})\big)\,. \end{aligned} \tag{42}$$

The result follows now from the linearity of the Jacobian. □

Now, we obtain for the gradient of the function f from (17), resp. (39):

$$\begin{aligned} \nabla f(\mathbf{z}) = \frac{1}{m}\Big(&\mathbf{A}(\mathbf{z})\mathbf{M} + \Big((\mathbf{Mz})^\top \otimes \mathbf{1}_{nm}\Big) D[\mathbf{A}](\mathbf{z}) \\ &- D[\mathbf{b}](\mathbf{z})\Big)^\top (\mathbf{A}(\mathbf{z})\mathbf{Mz} - \mathbf{b}(\mathbf{z}))\,. \end{aligned} \tag{43}$$

The addition of $\big((\mathbf{Mz})^\top \otimes \mathbf{1}_{nm}\big) D[\mathbf{A}](\mathbf{z}) - D[\mathbf{b}](\mathbf{z})$ stems from the inner derivative, since $\mathbf{A}$ and $\mathbf{b}$ are not constant.

4.4 Approximation of the gradient of f

Our numerical scheme depends on a gradient descent step of f from (17) (resp. (39)) with respect to $\mathbf{z}$. However, the evaluation of $\nabla f(\mathbf{z})$ is computationally expensive. It contains several matrix-matrix multiplications as well as the evaluation of a matrix Jacobian and a Kronecker product. These computations need to be done in every iteration. As we will see in Lemma 5, the evaluation of $\nabla f(\mathbf{z})$ can be done in a way, so that the main effort lies in computing n dyadics of vectors $\mathbf{S}\,[-\mathbf{M}_j\mathbf{z},1]^\top \in \mathbb{R}^{m,1}$ and $\left(\mathbf{M}_j^\top \mathbf{M}_j\mathbf{z}\right)^\top \in \mathbb{R}^{1,n}$.

In order to improve the performance of our numerical approach we further seek an approximation to ∇f that requires significantly less floating point operations. To this end, we assume for a moment that neither our matrix $\mathbf{A}$, nor our vector $\mathbf{b}$ depend on the unknown $\mathbf{z}$. In that case we obtain

$$\begin{aligned}\nabla f(\mathbf{z}) &= \nabla\left(\frac{1}{2m}\|\mathbf{AMz}-\mathbf{b}\|^2\right)\\ &= \frac{1}{m}(\mathbf{AM})^\top(\mathbf{AMz}-\mathbf{b}) =: \mathbf{q}\,.\end{aligned} \tag{44}$$

Our conclusions from (44) are twofold. First of all, $-\mathbf{q}$ seems to be a good candidate for a descent direction. At least when our data $\mathbf{A}$ and $\mathbf{b}$ does not depend on $\mathbf{z}$, then $-\mathbf{q}$ is an optimal and significantly easier to evaluate descent direction. Secondly, we could exploit (44) to derive an accelerated version of the iPiano algorithm for our task at hand. If we applied a lagged iteration on the descent step of f, then our matrix $\mathbf{A}$ and our vector $\mathbf{b}$ would become automatically independent of our current iterate and $-\mathbf{q}$ would be the steepest descent direction. The fact that $\mathbf{q}$ would not have to be recomputed in every iteration could outweigh the loss of accuracy and yield an additional performance boost.

The following theorem states precise conditions under which the vector $-\mathbf{q}$, defined in (44), yields a descent direction. Let us emphasise that Theorem 2 even allows a dependency on $\mathbf{z}$ in $\mathbf{A}$ and $\mathbf{b}$.

Theorem 2 *The vector*

$$-\mathbf{q} \coloneqq -\frac{1}{m}(\mathbf{A}(\mathbf{z})\mathbf{M})^\top(\mathbf{A}(\mathbf{z})\mathbf{Mz}-\mathbf{b}(\mathbf{z})) \tag{45}$$

is a descent direction for $f(\mathbf{z})$ from (17) *(resp.* (39)*) at position $\mathbf{z}$ if the expression*

$$\begin{aligned}&\langle(\mathbf{A}(\mathbf{z})\mathbf{Mz}-\mathbf{b}(\mathbf{z})),\\ &\qquad \mathbf{A}(\mathbf{z})\mathbf{M}D[\mathbf{AMz}-\mathbf{b}]^\top(\mathbf{A}(\mathbf{z})\mathbf{Mz}-\mathbf{b}(\mathbf{z}))\rangle\end{aligned} \tag{46}$$

is non-negative. This follows in particular, if $\mathbf{A}(\mathbf{z})\mathbf{M}D[\mathbf{AMz}-\mathbf{b}]^\top$ is positive semi-definite.

Proof Reordering the terms for $\nabla f(\mathbf{z})$ in (43) *yields the following relation between $\nabla f(\mathbf{z})$ and $\mathbf{q}$*

$$\begin{aligned}&\nabla f(\mathbf{z})\\ &\quad= \mathbf{q}+\frac{1}{m}\underbrace{\left(((\mathbf{Mz})^\top\otimes\mathbf{1})D[\mathbf{A}]-D[\mathbf{b}]\right)^\top}_{=D[\mathbf{AMz}-\mathbf{b}]-\mathbf{AM}}(\mathbf{AMz}-\mathbf{b})\,,\end{aligned} \tag{47}$$

where we have omitted the obvious dependencies on $\mathbf{z}$. Our vector $-\mathbf{q}$ will be a descent direction if $\langle-\mathbf{q},\nabla f\rangle \le 0$. Using (47) *we conclude*

$$\begin{aligned}\langle\mathbf{q},\nabla f\rangle &= \langle\mathbf{q},\mathbf{q}\rangle\\ &\quad+\frac{1}{m}\langle\mathbf{q},(D[\mathbf{AMz}-\mathbf{b}]-\mathbf{AM})^\top(\mathbf{AMz}-\mathbf{b})\rangle\,.\end{aligned} \tag{48}$$

Expanding $\mathbf{q}$ in the second inner product yields

$$\begin{aligned}&\langle\mathbf{q},(D[\mathbf{AMz}-\mathbf{b}]-\mathbf{AM})^\top(\mathbf{AMz}-\mathbf{b})\rangle\\ &=\langle(\mathbf{AM})^\top(\mathbf{AMz}-\mathbf{b}),\\ &\quad(D[\mathbf{AMz}-\mathbf{b}]-\mathbf{AM})^\top(\mathbf{AMz}-\mathbf{b})\rangle\\ &=\langle(\mathbf{AM})^\top(\mathbf{AMz}-\mathbf{b}),\\ &\quad D[\mathbf{AMz}-\mathbf{b}]^\top(\mathbf{AMz}-\mathbf{b})\rangle-\langle\mathbf{q},\mathbf{q}\rangle\,.\end{aligned} \tag{49}$$

Thus, we obtain

$$\begin{aligned}&\langle\mathbf{q},\nabla f\rangle\\ &=\frac{1}{m}\langle(\mathbf{AM})^\top(\mathbf{AMz}-\mathbf{b}),D[\mathbf{AMz}-\mathbf{b}]^\top(\mathbf{AMz}-\mathbf{b})\rangle\\ &=\frac{1}{m}\langle(\mathbf{AMz}-\mathbf{b}),(\mathbf{AM})D[\mathbf{AMz}-\mathbf{b}]^\top(\mathbf{AMz}-\mathbf{b})\rangle\,.\end{aligned} \tag{50}$$

Now, we are in presence of a descent direction whenever the expression

$$\langle(\mathbf{AMz}-\mathbf{b}),\mathbf{AM}D[\mathbf{AMz}-\mathbf{b}]^\top(\mathbf{AMz}-\mathbf{b})\rangle \tag{51}$$

is non-negative. This follows in particular, if the matrix

$$\mathbf{A}(\mathbf{z})\mathbf{M}D[\mathbf{AMz}-\mathbf{b}(\mathbf{z})]^\top \tag{52}$$

is positive semi-definite. □

The matrix $\mathbf{A}(\mathbf{z})\mathbf{M}D[\mathbf{AMz}-\mathbf{b}]^\top$ being positive semi-definite is a sufficient condition for our approximation $-\mathbf{q}$ being a descent direction with respect to f, which in turn is necessary for the convergence of the iPiano algorithm.

Let us conclude this section by remarking that the matrix

$$\mathbf{A}(\mathbf{z})\mathbf{M}D[\mathbf{AMz}-\mathbf{b}(\mathbf{z})]^\top \tag{53}$$

does not have any particular structure. Indeed, in general, it is made up from a product of non-symmetric and non-square matrices. Thus, additional claims on the spectral properties of this matrix are difficult to derive.

Nevertheless, we conjecture that

$$\nabla f(\mathbf{z}) \approx \frac{1}{m}\left(\mathbf{A}(\mathbf{z})\mathbf{M}\right)^\top\left(\mathbf{A}(\mathbf{z})\mathbf{Mz}-\mathbf{b}(\mathbf{z})\right) \tag{54}$$

is an efficient way to approximate $\nabla f(\mathbf{z})$ for computations. We will investigate possible deficiencies later in our numerical experiments.

Theorem 2 may be of use for enhancements of the proposed method, *e.g.* for designing break criteria of the inner loop of Algorithm 1 or for restart or kicking approaches.

4.5 Summary of the solution strategy

Our final algorithm for the computation of the depth and the albedo is given in Algorithm 1. For the step sizes we employed the "lazy backtracking" algorithm as in [25]. This includes increasing the Lipschitz constant $L^{(\ell)}$ for ∇f by multiplication with a parameter $\eta > 1$ ($\eta = 1.2$ in our experiments), until the new iterate $\tilde{\mathbf{z}}^{(l+1)}$ fulfils

$$f(\tilde{\mathbf{z}}^{(\ell+1)}) \leq f(\tilde{\mathbf{z}}^{(\ell)}) + \langle \nabla f(\tilde{\mathbf{z}}^{l}), \tilde{\mathbf{z}}^{(\ell+1)} - \tilde{\mathbf{z}}^{(\ell)} \rangle + \frac{L^{(\ell)}}{2} \left\| \tilde{\mathbf{z}}^{(\ell+1)} - \tilde{\mathbf{z}}^{(\ell)} \right\|_2^2 . \quad (55)$$

The found Lipschitz constant $L^{(\ell)}$ divided by a $\mu \geq 1$ ($\mu = 1.05$ in our experiments) delivers the start for estimating the Lipschitz constant $L^{(\ell+1)}$ in the next iPiano iteration.

Algorithm 1: Inertial Proximal Point Algorithm for Photometric Stereo

Choose prior $\mathbf{z}_0$ (classic PS), prior weight λ (10^{-6}), $c > 0$ (0.01) and $d > c$ (1)
Initialise $\mathbf{z}^{(0)}$ ($\mathbf{z}_0$) and $\rho^{(0)}$ (classic PS), and set $k = 0$
repeat
  Set $\tilde{\mathbf{z}}^{(0)} = \tilde{\mathbf{z}}^{(-1)} = \mathbf{z}^{(k)}$, $\delta^{(-1)} = d$ and $\ell = 0$
  repeat
    Lipschitz constant $L^{(\ell)}$ estimation by lazy backtracking
    Aux. variable: $\nu = \dfrac{\delta^{(\ell-1)} + L^{(\ell)}/2}{c + L^{(\ell)}/2}$
    Step size updates: $\beta^{(\ell)} = \dfrac{\nu - 1}{\nu + c - 0.5}$ and $\alpha^{(\ell)} = \dfrac{1-\beta}{c + L^{(\ell)}/2}$
    Aux. variable: $\delta^{(\ell)} = \dfrac{1}{\alpha^{(\ell)}} - \dfrac{L^{(\ell)}}{2} - \dfrac{\beta^{(\ell)}}{\alpha^{(\ell)}}$
    Depth update: $\tilde{\mathbf{z}}^{(\ell+1)} = \operatorname{prox}_{\alpha^{(\ell)} g}\left(\tilde{\mathbf{z}}^{(\ell)} - \alpha^{(\ell)} \nabla f\left(\tilde{\mathbf{z}}^{(\ell)}\right) + \beta\left(\tilde{\mathbf{z}}^{(\ell)} - \tilde{\mathbf{z}}^{(\ell-1)}\right)\right)$
    $\ell = \ell + 1$
  until *iPiano convergence*
  $\mathbf{z}^{(k+1)} = \tilde{\mathbf{z}}^{(\ell+1)}$
  Albedo update using (22)
  $k = k + 1$
until *global convergence*

In Algorithm 1 we could also use a constant step size $\beta \in [0, 1[$, so that the computation of ν and $\delta^{(\ell)}$ would not be required. By using $\beta = 0.5$ in our numerical experiments we achieved comparable results with respect to both computation time and quality of the reconstructed surface. However, by applying a variable step size $\beta^{(\ell)}$ deduced from the proof of Lemma 4.6 in [25] we ensure that the auxiliary sequence $\{\delta^{(\ell)}\}_{\ell=-1}^{\infty}$ is monotonically decreasing and therefore the convergence theory provided in [25] can be applied. To this end, let us remark that $\|\mathbf{z}\|_2 \to \infty$ implies $g(\mathbf{z}) \to \infty$ and that f is non-negative. Thus, $f + g$ is coercive. Furthermore g is convex and non-negative, such that $f + g$ is bounded below. The function f is obviously differentiable, *c.f.* (43).

The final ingredient to apply the general convergence result is the Lipschitz continuity of ∇f, which we will investigate in the following section. As a motivation, we repeat the general convergence result that was provided in [25], Theorem 4.8. For the definition of Lipschitz continuity we refer to (58).

Proposition 1 *Let $\{\tilde{\mathbf{z}}^{(\ell)}\}_{\ell=0}^{\infty}$ be a sequence generated by the inner loop of Algorithm 1, with ∇f computed according to* (43)*. If ∇f is Lipschitz continuous, then the following properties hold:*

1. *The sequence $\{f(\tilde{\mathbf{z}}^{(\ell)}) + g(\tilde{\mathbf{z}}^{(\ell)})\}_{\ell=0}^{\infty}$ converges.*
2. *There exists a converging subsequence $\{\tilde{\mathbf{z}}^{(\ell_i)}\}_{i=0}^{\infty}$.*
3. *For any limit point $\tilde{\mathbf{z}}^* := \lim_{i\to\infty} \tilde{\mathbf{z}}^{(\ell_i)}$ we have*

$$0 = \nabla f(\tilde{\mathbf{z}}^*) + \nabla g(\tilde{\mathbf{z}}^*) \quad (56)$$

and

$$\lim_{i\to\infty} f\left(\tilde{\mathbf{z}}^{(\ell_i)}\right) + g\left(\tilde{\mathbf{z}}^{(\ell_i)}\right) = f(\tilde{\mathbf{z}}^*) + g(\tilde{\mathbf{z}}^*) . \quad (57)$$

5 Convergence analysis

In Algorithm 1, the Lipschitz constant L of $\nabla f(\tilde{\mathbf{z}})^{(\ell)}$ is estimated by a lazy backtracking strategy. To derive a Lipschitz estimate for $\nabla f(\mathbf{z})$ for all $\mathbf{z} \in \mathbb{R}^n$ and thereby ensure that the convergence theory for the iPiano algorithm provided in [24, 25] can be applied, we first recall some general techniques to combine Lipschitz estimates. Afterwards, with these techniques we derive Lipschitz estimates for the gradient of f as well as for our approximation of this gradient.

The proofs in Sections 5.1 and 5.2 are lengthy and technical. The main result of these paragraphs consists of Proposition 2 followed by further explanations.

We conclude the convergence analysis by highlighting some aspects of the iPiano method. Thereby, we further justify our choice of a non-constant stepsize $\beta^{(\ell)}$, which might seem as a technical complication at first glance.

5.1 Technical Preliminaries

Although the final Lipschitz estimates for ∇f and $\mathbf{q}$, that we are interested in, involves a vector valued function with a

vector valued input, to get there we will in the most general case discuss Lipschitz estimates for $\mathbf{F}:\mathbb{R}^p \to \mathbb{R}^{q,r}$ with different choices for $\mathbf{F}$, p, q and r. The function $\mathbf{F}$ is Lipschitz continuous with a Lipschitz constant $L^{\mathbf{F}}$, if

$$\|\mathbf{F}(\mathbf{x})-\mathbf{F}(\mathbf{y})\|_2 \leq L^{\mathbf{F}}\|\mathbf{x}-\mathbf{y}\|_2 \qquad \text{for all } \mathbf{x},\mathbf{y}\in\mathbb{R}^p\,. \tag{58}$$

The following lemma contains some basic techniques to combine Lipschitz estimates. The proof is included for convenience.

Lemma 3 *Let $\mathbf{F}_1:\mathbb{R}^m\to\mathbb{R}^{n,p}$ and $\mathbf{F}_2:\mathbb{R}^m\to\mathbb{R}^{p,q}$ be Lipschitz continuous with $L^{(1)},L^{(2)}>0$, such that*

$$\|\mathbf{F}_k(\mathbf{x})-\mathbf{F}_k(\mathbf{y})\|_2 \leq L^{(k)}\|\mathbf{x}-\mathbf{y}\|_2\,, \tag{59}$$

for all $\mathbf{x},\mathbf{y}\in\mathbb{R}^m$ and $k\in\{1,2\}$, then we have the following properties:

1. *If there exist constants c_1,c_2, such that $\|\mathbf{F}_k(\mathbf{x})\|_2\leq c_k$ for all $\mathbf{x}\in\mathbb{R}^m$ and $k\in\{1,2\}$, then*

$$\begin{aligned}&\|\mathbf{F}_1(\mathbf{x})\mathbf{F}_2(\mathbf{x})-\mathbf{F}_1(\mathbf{y})\mathbf{F}_2(\mathbf{y})\|_2\\ &\qquad\leq\left(c_2L^{(1)}+c_1L^{(2)}\right)\|\mathbf{x}-\mathbf{y}\|_2\,.\end{aligned} \tag{60}$$

The following properties only concern the scalar case.

2. *If $n=p=1$ and if there exists a constant $c_3>0$, such that $\mathbf{F}_1(\mathbf{x})\geq c_3$ for all $\mathbf{x}\in\mathbb{R}^m$, then*

$$\left|\sqrt{\mathbf{F}_1(\mathbf{x})}-\sqrt{\mathbf{F}_1(\mathbf{y})}\right|\leq\frac{L^{(1)}}{2\sqrt{c_3}}\|\mathbf{x}-\mathbf{y}\|_2\,. \tag{61}$$

3. *If $n=p=1$ and if there exists a constant $c_4>0$, such that $|\mathbf{F}_1(\mathbf{x})|\geq c_4$ for all $\mathbf{x}\in\mathbb{R}^m$, then*

$$\left|\frac{1}{\mathbf{F}_1(\mathbf{x})}-\frac{1}{\mathbf{F}_1(\mathbf{y})}\right|\leq\frac{L^{(1)}}{(c_4)^2}\|\mathbf{x}-\mathbf{y}\|_2\,. \tag{62}$$

Proof *Let $\mathbf{x},\mathbf{y}\in\mathbb{R}^m$.*

1. *Since*

$$\begin{aligned}&\|\mathbf{F}_1(\mathbf{x})\mathbf{F}_2(\mathbf{x})-\mathbf{F}_1(\mathbf{y})\mathbf{F}_2(\mathbf{y})\|_2\\ &=\|\mathbf{F}_1(\mathbf{x})\mathbf{F}_2(\mathbf{x})-\mathbf{F}_1(\mathbf{x})\mathbf{F}_2(\mathbf{y})\\ &\quad+\mathbf{F}_1(\mathbf{x})\mathbf{F}_2(\mathbf{y})-\mathbf{F}_1(\mathbf{y})\mathbf{F}_2(\mathbf{y})\|_2\\ &\leq\|\mathbf{F}_1(\mathbf{x})\|_2\|\mathbf{F}_2(\mathbf{x})-\mathbf{F}_2(\mathbf{y})\|_2\\ &\quad+\|\mathbf{F}_2(\mathbf{y})\|_2\|\mathbf{F}_1(\mathbf{x})-\mathbf{F}_1(\mathbf{y})\|_2\,,\end{aligned} \tag{63}$$

   by $\|\mathbf{F}_k(\mathbf{x})\|_2\leq c_k$ for all $\mathbf{x}\in\mathbb{R}^m$ and $k\in\{1,2\}$ and (59) *we get* (60).
2. *Now let $n=p=1$.*
   If we have $\mathbf{F}_1(\mathbf{x})\geq c_3>0$ for all $\mathbf{x}\in\mathbb{R}^m$, then

$$\begin{aligned}&\left|\sqrt{\mathbf{F}_1(\mathbf{x})}-\sqrt{\mathbf{F}_1(\mathbf{y})}\right|=\left|\frac{\mathbf{F}_1(\mathbf{x})-\mathbf{F}_1(\mathbf{y})}{\sqrt{\mathbf{F}_1(\mathbf{x})}+\sqrt{\mathbf{F}_1(\mathbf{y})}}\right|\\ &\leq\left|\frac{1}{\sqrt{\mathbf{F}_1(\mathbf{x})}+\sqrt{\mathbf{F}_1(\mathbf{y})}}\right||\mathbf{F}_1(\mathbf{x})-\mathbf{F}_1(\mathbf{y})|\end{aligned} \tag{64}$$

   and therefore, with (59) *we get* (61).
3. *If we have $|\mathbf{F}_1(\mathbf{x})|\geq c_4>0$ for all $\mathbf{x}\in\mathbb{R}^m$, then*

$$\begin{aligned}&\left|\frac{1}{\mathbf{F}_1(\mathbf{x})}-\frac{1}{\mathbf{F}_1(\mathbf{y})}\right|=\left|\frac{\mathbf{F}_1(\mathbf{y})-\mathbf{F}_1(\mathbf{x})}{\mathbf{F}_1(\mathbf{x})\mathbf{F}_1(\mathbf{y})}\right|\\ &\leq\left|\frac{1}{\mathbf{F}_1(\mathbf{x})\mathbf{F}_1(\mathbf{y})}\right||\mathbf{F}_1(\mathbf{x})-\mathbf{F}_1(\mathbf{y})|\end{aligned} \tag{65}$$

   and with (59) *we get* (62). □

5.2 Lipschitz constant for the gradient of f

In this section we investigate the existence of a finite Lipschitz constant $L^{\nabla f}$, such that for all $\mathbf{x}$, $\mathbf{y}\in\mathbb{R}^n$

$$\|\nabla f(\mathbf{x})-\nabla f(\mathbf{y})\|_2\leq L^{\nabla f}\|\mathbf{x}-\mathbf{y}\|_2\,. \tag{66}$$

We will also investigate the existence of a Lipschitz constant $L^{\mathbf{q}}$ of the approximated gradient $\mathbf{q}$ from (44), as well as the dependencies of $L^{\nabla f}$ and $L^{\mathbf{q}}$ on n and m.

We make the following assumptions:

(A1) For all $j\in\{1,\dots,n\}$ the approximation of the spatial gradient $\mathbf{M}_j\mathbf{z}$ is bounded, *i.e.* there is a $L_j^{\mathbf{z}}\in[0,\infty)$, such that $\left\|\mathbf{M}_j\mathbf{z}\right\|_2\leq L_j^{\mathbf{z}}$ for all $\mathbf{z}\in\mathbb{R}^n$.

(A2) For $n\to\infty$ we have $L_j^{\mathbf{z}}\to 0$ for all $j\in\{1,\dots,n\}$, notably $L_j^{\mathbf{z}}\in O(1/\sqrt{n})$.

Let us remark that the previous assumptions on the decrease rate are done under the assumption, that the grid step size of our image remains the same when the number of pixels increases. While the finiteness of $L^{\nabla f}$ and $L^{\mathbf{q}}$ hinges on (A1), assumption (A2) is only needed to derive the dependencies of the Lipschitz constants on n. Although these are fairly strong assumptions, we choose not to switch to a more restricted space than $\mathbb{R}^n$, and instead assume that the depth map that is to be reconstructed and also the iterates $\tilde{\mathbf{z}}^{(\ell)}$ in Algorithm 1 fulfil (A1) and (A2).

If (A1) holds true, then we additionally define

$$\tilde{L}_j^{\mathbf{z}}\coloneqq\sqrt{1+\left(L_j^{\mathbf{z}}\right)^2}\qquad\text{for all } j\in\{1,\dots,n\}\,, \tag{67}$$

so that we have

$$\sqrt{1+\left\|\mathbf{M}_j\mathbf{z}\right\|_2^2}\leq\tilde{L}_j^{\mathbf{z}}\qquad\text{for all } \mathbf{z}\in\mathbb{R}^n\,. \tag{68}$$

If (A2) holds true, then $\tilde{L}_j^{\mathbf{z}}\in O(1)$ for all $j\in\{1,\dots,n\}$.

To obtain a Lipschitz estimate of the gradient

$$\nabla f(\mathbf{z})=\frac{1}{m}\left(\left((\mathbf{M}\mathbf{z})^\top\otimes\mathbf{1}\right)D[\mathbf{A}]+\mathbf{A}\mathbf{M}-D[\mathbf{b}]\right)^\top(\mathbf{A}\mathbf{M}\mathbf{z}-\mathbf{b}) \tag{69}$$

we will first derive Lipschitz estimates for the individual components and then combine them by using Lemma 3.

Corollary 3 *Let $j \in \{1,\dots,n\}$ and $\mathbf{x},\mathbf{y} \in \mathbb{R}^n$. In addition, we define $\kappa := L_j^{\mathbf{z}} \left\|\mathbf{M}_j\right\|$. If (A1) holds true, then*

$$\left| \left\|\mathbf{M}_j\mathbf{x}\right\|_2^2 - \left\|\mathbf{M}_j\mathbf{y}\right\|_2^2 \right| \leq 2\kappa \|\mathbf{x}-\mathbf{y}\|_2 \,, \tag{70}$$

$$\left| \sqrt{1+\left\|\mathbf{M}_j\mathbf{x}\right\|_2^2} - \sqrt{1+\left\|\mathbf{M}_j\mathbf{y}\right\|_2^2} \right| \leq \kappa \|\mathbf{x}-\mathbf{y}\|_2 \,, \tag{71}$$

$$\left| \frac{1}{\sqrt{1+\left\|\mathbf{M}_j\mathbf{x}\right\|_2^2}} - \frac{1}{\sqrt{1+\left\|\mathbf{M}_j\mathbf{y}\right\|_2^2}} \right| \leq \kappa \|\mathbf{x}-\mathbf{y}\|_2 \,, \tag{72}$$

$$\left| \frac{1}{1+\left\|\mathbf{M}_j\mathbf{x}\right\|_2^2} - \frac{1}{1+\left\|\mathbf{M}_j\mathbf{y}\right\|_2^2} \right| \leq 2\kappa \|\mathbf{x}-\mathbf{y}\|_2 \,, \tag{73}$$

$$\left| \frac{1}{\sqrt{1+\left\|\mathbf{M}_j\mathbf{x}\right\|_2^2}^3} - \frac{1}{\sqrt{1+\left\|\mathbf{M}_j\mathbf{y}\right\|_2^2}^3} \right| \leq 3\kappa \|\mathbf{x}-\mathbf{y}\|_2 \,, \tag{74}$$

$$\begin{aligned} &\left\| \begin{bmatrix} -\mathbf{M}_j\mathbf{x} \\ 1 \end{bmatrix} (\mathbf{M}_j\mathbf{x})^\top - \begin{bmatrix} -\mathbf{M}_j\mathbf{y} \\ 1 \end{bmatrix} (\mathbf{M}_j\mathbf{y})^\top \right\|_2 \\ &\qquad \leq \left(\tilde{L}_j^{\mathbf{z}} + L_j^{\mathbf{z}}\right) \left\|\mathbf{M}_j\right\| \|\mathbf{x}-\mathbf{y}\|_2 \,. \end{aligned} \tag{75}$$

Proof Using Lemma 3.1 with $\mathbf{F}_1(\mathbf{z}) := (\mathbf{M}_j\mathbf{z})^\top$, $\mathbf{F}_2(\mathbf{z}) := \mathbf{M}_j\mathbf{z}$, (A1) and $\left\|\mathbf{M}_j\mathbf{x}-\mathbf{M}_j\mathbf{y}\right\|_2 \leq \left\|\mathbf{M}_j\right\| \|\mathbf{x}-\mathbf{y}\|_2$ we can deduce (70).

With Lemma 3.2, $\mathbf{F}_1(\mathbf{z}) := 1+\left\|\mathbf{M}_j\mathbf{z}\right\|_2^2 \geq 1$ for all $\mathbf{z} \in \mathbb{R}^n$ and the just shown validity of (70) *we get* (71).

Making use of Lemma 3.3, $\mathbf{F}_1(\mathbf{z}) := \sqrt{1+\left\|\mathbf{M}_j\mathbf{z}\right\|_2^2} \geq 1$ holds for all $\mathbf{z} \in \mathbb{R}^n$ and by employing (71) *we obtain* (72).

By using Lemma 3.3, $\mathbf{F}_1(\mathbf{z}) := 1+\left\|\mathbf{M}_j\mathbf{z}\right\|_2^2 \geq 1$ for all $\mathbf{z} \in \mathbb{R}^n$, and together with (70) *we get* (73).

By using Lemma 3.1, $\mathbf{F}_1(\mathbf{z}) := 1/\sqrt{1+\left\|\mathbf{M}_j\mathbf{z}\right\|_2^2} \leq 1$ and $\mathbf{F}_2(\mathbf{z}) := 1/\left(1+\left\|\mathbf{M}_j\mathbf{z}\right\|_2^2\right) \leq 1$ for all $\mathbf{z} \in \mathbb{R}^n$, and together with (72) *and* (73) *it is easy to see that* (74) *is true.*

By combining Lemma 3.1, $\mathbf{F}_1(\mathbf{z}) := [-\mathbf{M}_j\mathbf{z},1]^\top$, $\mathbf{F}_2(\mathbf{z}) := (\mathbf{M}_j\mathbf{z})^\top$, (A1), (68) *and $\left\|\mathbf{M}_j\mathbf{x}-\mathbf{M}_j\mathbf{y}\right\|_2 \leq \left\|\mathbf{M}_j\right\| \|\mathbf{x}-\mathbf{y}\|_2$ we finally get the validity of* (75). □

The following lemma contains the first indication of Lipschitz estimates.

Lemma 4 *Let $\mathbf{x},\mathbf{y} \in \mathbb{R}^n$ and $\mathbf{A}$ be defined as in* (15). *If (A1) holds true, then*

$$\left\|\mathbf{A}_j(\mathbf{x})-\mathbf{A}_j(\mathbf{y})\right\|_2 \leq \underbrace{\rho_j \|\mathbf{S}_\ell\|_2 L_j^{\mathbf{z}} \left\|\mathbf{M}_j\right\|}_{=:L_j^{\mathbf{A}}} \|\mathbf{x}-\mathbf{y}\|_2 \tag{76}$$

for all $j \in \{1,\dots,n\}$ as well as

$$\|\mathbf{A}(\mathbf{x})-\mathbf{A}(\mathbf{y})\|_2 \leq \underbrace{\left(\max_j L_j^{\mathbf{A}}\right)}_{=:L^{\mathbf{A}}} \|\mathbf{x}-\mathbf{y}\|_2 \,. \tag{77}$$

If additionally (A2) holds true, then

$$L_j^{\mathbf{A}} \in O(\sqrt{m/n}) \qquad \text{for all } j \in \{1,\dots,n\} \,, \tag{78}$$

$$L^{\mathbf{A}} \in O(\sqrt{m/n}) \,. \tag{79}$$

Proof Let $j \in \{1,\dots,n\}$. From the definition of $\mathbf{A}_j$ in (13) *and from* (72) *follows directly* (76).

By $\rho_j \in [0,1]$, $\|\mathbf{S}_\ell\|_2 \in O(\sqrt{m})$, (A2) and $\left\|\mathbf{M}_j\right\| \in O(1)$ we obtain (78).

Since $\mathbf{A}$ is a (in general non-square) block diagonal matrix we have

$$\mathbf{A}^\top\mathbf{A} = \begin{bmatrix} \mathbf{A}_1^\top\mathbf{A}_1 & & \\ & \ddots & \\ & & \mathbf{A}_n^\top\mathbf{A}_n \end{bmatrix} \in \mathbb{R}^{2n,2n} \,, \tag{80}$$

and with

$$\begin{aligned} &\det\left(\mathbf{A}^\top\mathbf{A}-\lambda\mathbf{1}_{2n}\right) \\ &\qquad = \det\left(\mathbf{A}_1^\top\mathbf{A}_1-\lambda\mathbf{1}_2\right) \dots \det\left(\mathbf{A}_n^\top\mathbf{A}_n-\lambda\mathbf{1}_2\right) \end{aligned} \tag{81}$$

for all $\lambda \in \mathbb{R}$ we have

$$\begin{aligned} &\|\mathbf{A}\|_2 \\ &= \sqrt{\max\left\{\operatorname{eig}\left(\mathbf{A}^\top\mathbf{A}\right)\right\}} \\ &= \sqrt{\max\left\{\operatorname{eig}\left(\mathbf{A}_1^\top\mathbf{A}_1\right),\dots,\operatorname{eig}\left(\mathbf{A}_n^\top\mathbf{A}_n\right)\right\}} \\ &= \max_j \sqrt{\max\left\{\operatorname{eig}\left(\mathbf{A}_j^\top\mathbf{A}_j\right)\right\}} = \max_j \left\|\mathbf{A}_j\right\|_2 \,. \end{aligned} \tag{82}$$

In the same way we can derive the equation

$$\|\mathbf{A}(\mathbf{x})-\mathbf{A}(\mathbf{y})\|_2 = \max_j \left\|\mathbf{A}_j(\mathbf{x})-\mathbf{A}_j(\mathbf{y})\right\|_2 \,, \tag{83}$$

and with (76) *we obtain* (77).

From (77) *and* (78) *follows* (79). □

The following assertion is an immediate consequence of (77).

Corollary 4 *Let $\mathbf{x},\mathbf{y} \in \mathbb{R}^n$. If (A1) holds true, then*

$$\|\mathbf{A}(\mathbf{x})\mathbf{M}-\mathbf{A}(\mathbf{y})\mathbf{M}\|_2 \leq L^{\mathbf{A}} \|\mathbf{M}\|_2 \|\mathbf{x}-\mathbf{y}\|_2 \,. \tag{84}$$

We proceed with another building block used for coming to Proposition 2.

Corollary 5 *Let $\mathbf{x},\mathbf{y} \in \mathbb{R}^n$. If (A1) holds true, then*

$$\begin{aligned} &\left\|\mathbf{A}_j(\mathbf{x})\mathbf{M}_j\mathbf{x}-\mathbf{b}_j(\mathbf{x})-\mathbf{A}_j(\mathbf{y})\mathbf{M}_j\mathbf{y}+\mathbf{b}_j(\mathbf{y})\right\|_2 \\ &\qquad \leq \underbrace{\rho_j \|\mathbf{S}\|_2 \left\|\mathbf{M}_j\right\| \left(\tilde{L}_j^{\mathbf{z}} L_j^{\mathbf{z}}+1\right)}_{=:L_j^f} \|\mathbf{x}-\mathbf{y}\|_2 \end{aligned} \tag{85}$$

for all $j \in \{1,\dots,n\}$ *and*

$$\|\mathbf{A}(\mathbf{x})\mathbf{M}\mathbf{x}-\mathbf{b}(\mathbf{x})-\mathbf{A}(\mathbf{y})\mathbf{M}\mathbf{y}+\mathbf{b}(\mathbf{y})\|_2 \leq \sqrt{\sum_{j=1}^{n}\left(L_j^f\right)^2}\|\mathbf{x}-\mathbf{y}\|_2 =: L^f\|\mathbf{x}-\mathbf{y}\|_2 . \quad (86)$$

If additionally (A2) holds true, then for all $j \in \{1,\dots,n\}$

$$L_j^f \in O(\sqrt{m}) , \qquad L^f \in O(\sqrt{mn}) . \quad (87)$$

Proof From Lemma 3.1 with

$$\mathbf{F}_1(\mathbf{z}) \coloneqq \frac{\rho_j}{\sqrt{1+\left\|\mathbf{M}_j\mathbf{z}\right\|}}\mathbf{S} , \qquad \mathbf{F}_2(\mathbf{z}) \coloneqq \begin{bmatrix}-\mathbf{M}_j\mathbf{z}\\ 1\end{bmatrix} , \quad (88)$$

$\|\mathbf{F}_1(\mathbf{z})\|_2 \leq \rho_j\|\mathbf{S}\|_2$, (68), (72) *and the Lipschitz estimate* $\|\mathbf{F}_2(\mathbf{x})-\mathbf{F}_2(\mathbf{x})\|_2 \leq \left\|\mathbf{M}_j\right\|\|\mathbf{x}-\mathbf{x}\|_2$ *follows* (85).

From (85) *and with the definition of the Euclidean norm we obtain* (86).

The inclusion (87) *follows from (A2),* $\rho_j \in O(1)$ *and* $\left\|\mathbf{M}_j\right\| \in O(1)$ *for all* $j \in \{1,\dots,n\}$ *and* $\|\mathbf{S}\|_2 \in O(\sqrt{m})$. □

The following lemma will subsequently be used to derive a Lipschitz estimate for ∇f, but it also shows a more explicit representation of the exact gradient.

Lemma 5 *For* $\mathbf{z} \in \mathbb{R}^n$ *we have*

$$\mathbf{p}(\mathbf{z}) \coloneqq \left((\mathbf{M}\mathbf{z})^\top \otimes \mathbf{1}_{mn}\right) D[\mathbf{A}](\mathbf{z}) - D[\mathbf{b}](\mathbf{z}) = \begin{bmatrix} -\frac{\rho_1}{\sqrt{1+\|\mathbf{M}_1\mathbf{z}\|_2^2}^3}\mathbf{S}\begin{bmatrix}-\mathbf{M}_1\mathbf{z}\\ 1\end{bmatrix}\left(\mathbf{M}_1^\top\mathbf{M}_1\mathbf{z}\right)\top \\ \vdots \\ -\frac{\rho_n}{\sqrt{1+\|\mathbf{M}_n\mathbf{z}\|_2^2}^3}\mathbf{S}\begin{bmatrix}-\mathbf{M}_n\mathbf{z}\\ 1\end{bmatrix}\left(\mathbf{M}_n^\top\mathbf{M}_n\mathbf{z}\right)^\top \end{bmatrix} . \quad (89)$$

Proof To find an expression for $\mathbf{p}$ *without the Kronecker product, we will simply write down all components of* $\mathbf{p}$ *and consecutively join them.*

For $\mathbf{z} \in \mathbb{R}^n$ *and* $j \in \{1,\dots,n\}$ *we have*

$$\mathbf{b}_j = \mathbf{I}_j - \frac{\rho_j}{\sqrt{1+\left\|\mathbf{M}_j\mathbf{z}\right\|_2^2}}\mathbf{S}_r , \quad (90)$$

$$D[\mathbf{b}_j](\mathbf{z}) = \frac{\rho_j}{\sqrt{1+\left\|\mathbf{M}_j\mathbf{z}\right\|_2^2}^3}\mathbf{S}_r\left(\mathbf{M}_j^\top\mathbf{M}_j\mathbf{z}\right)^\top \in \mathbb{R}^{m,n} , \quad (91)$$

leading to

$$D[\mathbf{b}](\mathbf{z}) = \begin{bmatrix} \frac{\rho_1}{\sqrt{1+\|\mathbf{M}_1\mathbf{z}\|_2^2}^3}\mathbf{S}_r\left(\mathbf{M}_1^\top\mathbf{M}_1\mathbf{z}\right)^\top \\ \vdots \\ \frac{\rho_n}{\sqrt{1+\|\mathbf{M}_n\mathbf{z}\|_2^2}^3}\mathbf{S}_r\left(\mathbf{M}_n^\top\mathbf{M}_n\mathbf{z}\right)^\top \end{bmatrix} \in \mathbb{R}^{mn,n} . \quad (92)$$

For $k \in \{1,2\}$ *with* $\mathbf{A}_j^k(\mathbf{z})$, $\mathbf{S}_\ell^k$ *and* $\mathbf{M}_j^k$ *we denote the k-th column of* $\mathbf{A}_j(\mathbf{z})$ *and* $\mathbf{S}_\ell$ *and the k-th row of* $\mathbf{M}_j$. *We derive*

$$D\left[\mathbf{A}_j^k\right](\mathbf{z}) = \frac{\rho_j\mathbf{S}_\ell^k}{\sqrt{1+\left\|\mathbf{M}_j\mathbf{z}\right\|_2^2}^3}\left(\mathbf{M}_j^\top\mathbf{M}_j\mathbf{z}\right)^\top \in \mathbb{R}^{m,n} . \quad (93)$$

Because of the structure of $\mathbf{A}$, *defined in* (15), *and our choice of the Jacobian matrix as per Definition 1, the Jacobian matrix of* $\mathbf{A}$ *has the form*

$$D[\mathbf{A}](\mathbf{z}) = \begin{bmatrix} D\left[\mathbf{A}_1^1\right](\mathbf{z}) \\ \mathbf{0}_{mn-m,n} \\ D\left[\mathbf{A}_1^2\right](\mathbf{z}) \\ \mathbf{0}_{mn,n} \\ D\left[\mathbf{A}_2^1\right](\mathbf{z}) \\ \mathbf{0}_{mn-m,n} \\ D\left[\mathbf{A}_2^2\right](\mathbf{z}) \\ \mathbf{0}_{mn,n} \\ \vdots \\ D\left[\mathbf{A}_n^1\right](\mathbf{z}) \\ \mathbf{0}_{mn-m,n} \\ D\left[\mathbf{A}_n^2\right](\mathbf{z}) \end{bmatrix} \in \mathbb{R}^{2mn^2,n} , \quad (94)$$

where $\mathbf{0}_{p,q}$ *is a* $p \times q$ *block of zeros. Since*

$$\mathbb{R}^{mn,2mn^2} \ni (\mathbf{M}\mathbf{z})^\top \otimes \mathbf{1}_{mn} = \begin{bmatrix} \mathbf{M}_1^1\mathbf{z} & & \mathbf{M}_1^2\mathbf{z} & & & \mathbf{M}_n^2\mathbf{z} \\ & \ddots & & \ddots & \cdots & & \ddots \\ & \mathbf{M}_1^1\mathbf{z} & & \mathbf{M}_1^2\mathbf{z} & & & \mathbf{M}_n^2\mathbf{z} \end{bmatrix} \quad (95)$$

we get

$$\begin{aligned}\left((\mathbf{M}\mathbf{z})^\top \otimes \mathbf{1}_{mn}\right) D[\mathbf{A}](\mathbf{z}) &= \begin{bmatrix} \mathbf{M}_1^1 D\left[\mathbf{A}_1^1\right](\mathbf{z}) + \mathbf{M}_1^2\mathbf{z}D\left[\mathbf{A}_1^2\right](\mathbf{z}) \\ \vdots \\ \mathbf{M}_n^1\mathbf{z}D\left[\mathbf{A}_n^1\right](\mathbf{z}) + \mathbf{M}_n^2\mathbf{z}D\left[\mathbf{A}_n^2\right](\mathbf{z}) \end{bmatrix} \\ &= \begin{bmatrix} \frac{\rho_1}{\sqrt{1+\|\mathbf{M}_1\mathbf{z}\|_2^2}^3}\mathbf{S}_\ell\mathbf{M}_1\mathbf{z}\left(\mathbf{M}_1^\top\mathbf{M}_1\mathbf{z}\right)\top \\ \vdots \\ \frac{\rho_n}{\sqrt{1+\|\mathbf{M}_n\mathbf{z}\|_2^2}^3}\mathbf{S}_\ell\mathbf{M}_n\mathbf{z}\left(\mathbf{M}_n^\top\mathbf{M}_n\mathbf{z}\right)^\top \end{bmatrix} .\end{aligned} \quad (96)$$

Therefore, together with (92), *we get*

$$\mathbf{p}(\mathbf{z}) = \begin{bmatrix} -\frac{\rho_1}{\sqrt{1+\|\mathbf{M}_1\mathbf{z}\|_2^2}^3}\mathbf{S}\begin{bmatrix}-\mathbf{M}_1\mathbf{z}\\ 1\end{bmatrix}\left(\mathbf{M}_1^\top\mathbf{M}_1\mathbf{z}\right)\top \\ \vdots \\ -\frac{\rho_n}{\sqrt{1+\|\mathbf{M}_n\mathbf{z}\|_2^2}^3}\mathbf{S}\begin{bmatrix}-\mathbf{M}_n\mathbf{z}\\ 1\end{bmatrix}\left(\mathbf{M}_n^\top\mathbf{M}_n\mathbf{z}\right)^\top \end{bmatrix} . \quad (97)$$

□

Corollary 6 *Let $\mathbf{z} \in \mathbb{R}^n$, $j \in \{1,\dots,n\}$ and*

$$\mathbf{p}_j(\mathbf{z}) := -\frac{\rho_j}{\sqrt{1+\left\|\mathbf{M}_j\mathbf{z}\right\|_2^2}^3}\mathbf{S}\begin{bmatrix}-\mathbf{M}_j\mathbf{z}\\ 1\end{bmatrix}\left(\mathbf{M}_j^\top\mathbf{M}_j\mathbf{z}\right)^\top . \quad (98)$$

If (A1) holds true, then

$$\left\|\mathbf{p}_j(\mathbf{x})-\mathbf{p}_j(\mathbf{y})\right\|_2 \leq \underbrace{\rho_j\|\mathbf{S}\|_2\left\|\mathbf{M}_j\right\|^2\left(3\tilde{L}_j^{\mathbf{z}}\left(L_j^{\mathbf{z}}\right)^2+\tilde{L}_j^{\mathbf{z}}+L_j^{\mathbf{z}}\right)}_{=:L_j^{\mathbf{p}}}\|\mathbf{x}-\mathbf{y}\|_2 . \quad (99)$$

If additionally (A2) holds true, then

$$L_j^{\mathbf{p}} \in O(\sqrt{m}) \quad (100)$$

Proof With Lemma 3.1 and the settings

$$\mathbf{F}_1(\mathbf{z}) := -\frac{\rho_j}{\sqrt{1+\left\|\mathbf{M}_j\mathbf{z}\right\|_2^2}^3}\mathbf{S}, \quad (101)$$

$$\mathbf{F}_2(\mathbf{z}) := [-\mathbf{M}_j\mathbf{z},1]^\top(\mathbf{M}_j\mathbf{z})^\top\mathbf{M}_j , \quad (102)$$

and with (74)*,* (75) *and (A1) we obtain* (99)*.*

Equation (100) *follows from $\|\mathbf{S}\|_2 \in O(\sqrt{m})$ and the estimate $\left(3\tilde{L}_j^{\mathbf{z}}\left(L_j^{\mathbf{z}}\right)^2+\tilde{L}_j^{\mathbf{z}}+L_j^{\mathbf{z}}\right) \in O(1)$ according to (A2).* □

Let us now present finally the main result of this section.

Proposition 2 *Let ∇f be defined as in* (43) *and $\mathbf{q}$ be defined as in* (44)*, $\mathbf{x},\mathbf{y} \in \mathbb{R}^n$. If (A1) holds true, then*

$$\|\nabla f(\mathbf{x})-\nabla f(\mathbf{y})\|_2 \leq L^{\nabla f}\|\mathbf{x}-\mathbf{y}\|_2 , \quad (103)$$

$$\|\mathbf{q}(\mathbf{x})-\mathbf{q}(\mathbf{y})\|_2 \leq L^{\mathbf{q}}\|\mathbf{x}-\mathbf{y}\|_2 , \quad (104)$$

where

$$\begin{aligned} L^{\nabla f} := \frac{1}{m}\Bigg(&\sqrt{\sum_{j=1}^n\left(\left\|\mathbf{I}_j\right\|_2^2+\rho_j^2\|\mathbf{S}\|_2^2\right)}\\ &\left(\sqrt{\sum_{j=1}^n\left(L_j^{\mathbf{p}}\right)^2}+L^{\mathbf{A}}\|\mathbf{M}\|_2\right)\\ &+L^f\Bigg(\max_j\rho_j\|\mathbf{S}_\ell\|_2\|\mathbf{M}\|_2\\ &+\sqrt{\sum_{j=1}^n\rho_j^2\|\mathbf{S}\|_2^2\left(\tilde{L}_j^{\mathbf{z}}L_j^{\mathbf{z}}\right)^2\left\|\mathbf{M}_j\right\|_2^2}\Bigg)\Bigg) , \end{aligned} \quad (105)$$

$$\begin{aligned} L^{\mathbf{q}} := \frac{1}{m}\Bigg(&\sqrt{\sum_{j=1}^n\left(\left\|\mathbf{I}_j\right\|_2^2+\rho_j^2\|\mathbf{S}\|_2^2\right)}L^{\mathbf{A}}\|\mathbf{M}\|_2\\ &+\max_j\rho_j\|\mathbf{S}_\ell\|_2\|\mathbf{M}\|_2L^f\Bigg) . \end{aligned} \quad (106)$$

If additionally (A2) holds true, then

$$L^{\nabla f} \in O(n) , \quad (107)$$

$$L^{\mathbf{q}} \in O(\sqrt{n}) . \quad (108)$$

Proof First we will derive a Lipschitz estimate for $\mathbf{q}$. Assume that (A1) holds true. We define

$$\mathbf{F}_1(\mathbf{z}) := \frac{1}{m}(\mathbf{A}(\mathbf{z})\mathbf{M})^\top , \quad (109)$$

$$\mathbf{F}_2(\mathbf{z}) := \mathbf{A}(\mathbf{z})\mathbf{M}\mathbf{z}-\mathbf{b}(\mathbf{z}) \quad (110)$$

for all $\mathbf{z} \in \mathbb{R}^n$. As in (82) *we get*

$$\|\mathbf{F}_1(\mathbf{z})\|_2 \leq \frac{1}{m}\max_j\rho_j\|\mathbf{S}_\ell\|_2\|\mathbf{M}\|_2 , \quad (111)$$

and we also have

$$\|\mathbf{F}_2(\mathbf{z})\|_2 \leq \sqrt{\sum_{j=1}^n\left(\left\|\mathbf{I}_j\right\|_2^2+\rho_j^2\|\mathbf{S}\|_2^2\right)} . \quad (112)$$

With Lemma 3.1 and the Lipschitz estimates (84) *and* (86) *we get* (104)*.*

To deduce the Lipschitz estimate for ∇f, we extend the proof by redefining

$$\mathbf{F}_1(\mathbf{z}) := \frac{1}{m}(\mathbf{A}(\mathbf{z})\mathbf{M}+\mathbf{p}(\mathbf{z}))^\top , \quad (113)$$

where $\mathbf{p}$ is defined as in (89)*. By* (99) *we get*

$$\|\mathbf{p}(\mathbf{x})-\mathbf{p}(\mathbf{y})\|_2 \leq \sqrt{\sum_{j=1}^n\left(L_j^{\mathbf{p}}\right)^2}\|\mathbf{x}-\mathbf{y}\|_2 . \quad (114)$$

Together with (84) *we obtain*

$$\begin{aligned} &\|\mathbf{F}_1(\mathbf{x})-\mathbf{F}_1(\mathbf{y})\|_2\\ &\qquad\leq \frac{1}{m}\left(L^{\mathbf{A}}\|\mathbf{M}\|_2+\sqrt{\sum_{j=1}^n\left(L_j^{\mathbf{p}}\right)^2}\right)\|\mathbf{x}-\mathbf{y}\|_2 . \end{aligned} \quad (115)$$

Furthermore by the definition of $\mathbf{p}$ in (89) *and analogously to* (111) *we get*

$$\begin{aligned} \|\mathbf{F}_1(\mathbf{z})\|_2 \leq \frac{1}{m}\Bigg(&\max_j\rho_j\|\mathbf{S}_\ell\|_2\|\mathbf{M}\|_2\\ &+\sqrt{\sum_{j=1}^n\rho_j^2\|\mathbf{S}\|_2^2\left(\tilde{L}_j^{\mathbf{z}}L_j^{\mathbf{z}}\right)^2\left\|\mathbf{M}_j\right\|_2^2}\Bigg) . \end{aligned} \quad (116)$$

Now by Lemma 3.1 with (86)*,* (112)*,* (115) *and* (116) *we can deduce* (103)*.*

Now assume that (A2) holds true. The inclusions (79)*, $\rho_j \in O(1)$, $\|\mathbf{M}\|_2 \in O(1)$, $\left\|\mathbf{I}_j\right\|_2 \in O(\sqrt{m})$ and $\|\mathbf{S}\|_2 \in O(\sqrt{m})$ lead to*

$$\sqrt{\sum_{j=1}^n\left(\left\|\mathbf{I}_j\right\|_2^2+\rho_j^2\|\mathbf{S}\|_2^2\right)}L^{\mathbf{A}}\|\mathbf{M}\|_2 \in O(m) . \quad (117)$$

Furthermore by (87)*, $\rho_j \in O(1)$, $\|\mathbf{M}\|_2 \in O(1)$ and $\|\mathbf{S}_\ell\|_2 \in O(\sqrt{m})$ we obtain*

$$\max_j\rho_j\|\mathbf{S}_\ell\|_2\|\mathbf{M}\|_2L^f \in O(m\sqrt{n}) . \quad (118)$$

Now from (117), (118) *and* $1/m \in O(1/m)$ *follows* (108).

The inclusions (100), $\rho_j \in O(1)$, $\|\mathbf{M}\|_2 \in O(1)$, $\left\|\mathbf{I}_j\right\|_2 \in O(\sqrt{m})$ *and* $\|\mathbf{S}\|_2 \in O(\sqrt{m})$ *lead to*

$$\sqrt{\sum_{j=1}^{n}\left(\left\|\mathbf{I}_j\right\|_2^2+\rho_j^2\|\mathbf{S}\|_2^2\right)}\sqrt{\sum_{j=1}^{n}\left(L_j^{\mathbf{p}}\right)^2} \in O(mn)\,. \tag{119}$$

By (87), $\rho_j \in O(1)$, $\left\|\mathbf{M}_j\right\| \in O(1)$, $\tilde{L}_j^{\mathbf{z}} \in O(1)$, $L_j^{\mathbf{z}} \in O(1/\sqrt{n})$ *and* $\|\mathbf{S}\|_2 \in O(\sqrt{m})$ *we obtain*

$$L^f\sqrt{\sum_{j=1}^{n}\rho_j^2\|\mathbf{S}\|_2^2\left(\tilde{L}_j^{\mathbf{z}}L_j^{\mathbf{z}}\right)^2\left\|\mathbf{M}_j\right\|^2} \in O(m)\,. \tag{120}$$

Finally from (117), (118), (119), (120) *and* $1/m \in O(1/m)$ *follows* (107). □

We have shown that under the assumptions (A1) and (A2) the gradient as well as the approximated gradient of f are Lipschitz continuous.

As already indicated in (55), for practical applications we may also be interested in local Lipschitz constants $L^{(\ell)}$ fulfilling

$$f(\tilde{\mathbf{z}}^{(\ell+1)}) \le f(\tilde{\mathbf{z}}^{(\ell)}) + \langle\nabla f(\tilde{\mathbf{z}}^{l}),\tilde{\mathbf{z}}^{(\ell+1)}-\tilde{\mathbf{z}}^{(\ell)}\rangle + \frac{L^{(\ell)}}{2}\left\|\tilde{\mathbf{z}}^{(\ell+1)}-\tilde{\mathbf{z}}^{(\ell)}\right\|_2^2, \tag{121}$$

following the "lazy backtracking" strategy as it was proposed for the iPiano algorithm in [25]. By testing for the validity of this inequality also very small Lipschitz constants may be accepted, if the new value $f(\tilde{\mathbf{z}}^{(\ell+1)})$ is even lower than what would be possible for a function f with an $L^{(\ell)}$-Lipschitz continuous gradient, for more details see also Section 6.1.

5.3 Descent properties of the iPiano algorithm

We have seen in Section 4.4 that it is not guaranteed that the approximated gradient $\mathbf{q}$ delivers a descent direction for the function $f(\mathbf{z})$. Testing if $-\mathbf{q}(\mathbf{z})$ is a descent direction could be done by computing the actual gradient $\nabla f(\mathbf{z})$, which is not desirable for practical applications.

Another test may be to watch for increasing energies $f(\tilde{\mathbf{z}}^{(\ell)})+g(\tilde{\mathbf{z}}^{(\ell)})$ during computations performed by the iPiano algorithm. However, iPiano does not enforce decreasing function values, but a descent property is given for a majorising sequence of values

$$H_{\delta^{(\ell)}}(\tilde{\mathbf{z}}^{(\ell)},\tilde{\mathbf{z}}^{(\ell-1)}) := f(\tilde{\mathbf{z}}^{(\ell)})+g(\tilde{\mathbf{z}}^{(\ell)})+\delta^{(\ell)}\Delta^{(\ell)}\,, \tag{122}$$

as pointed out in [25], Proposition 4.7, where

$$\Delta^{(\ell)} := \left\|\tilde{\mathbf{z}}^{(\ell)}-\tilde{\mathbf{z}}^{(\ell-1)}\right\|_2^2, \tag{123}$$

$$\delta^{(\ell)} := \frac{1}{\alpha^{(\ell)}}-\frac{L^{(\ell)}}{2}-\frac{\beta^{(\ell)}}{\alpha^{(\ell)}}\,. \tag{124}$$

For sequences $\{\tilde{\mathbf{z}}^{(\ell)}\}_{\ell=-1}^{\infty}$, $\{L^{(\ell)}\}_{\ell=0}^{\infty}$, $\{\alpha^{(\ell)}\}_{\ell=0}^{\infty}$ and $\{\beta^{(\ell)}\}_{\ell=0}^{\infty}$ generated by iPiano, the sequence $\{H_{\delta^{(\ell)}}(\tilde{\mathbf{z}}^{(\ell)},\tilde{\mathbf{z}}^{(\ell-1)})\}_{\ell=0}^{\infty}$ is monotonically decreasing, and for $\ell = 0,1,\ldots,$

$$H_{\delta^{(\ell+1)}}(\tilde{\mathbf{z}}^{(\ell+1)},\tilde{\mathbf{z}}^{(\ell)}) \le H_{\delta^{(\ell)}}(\tilde{\mathbf{z}}^{(\ell)},\tilde{\mathbf{z}}^{(\ell-1)})-\gamma^{(\ell)}\Delta^{(\ell)} \tag{125}$$

holds, where

$$\gamma^{(\ell)} := \frac{1}{\alpha^{(\ell)}}-\frac{L^{(\ell)}}{2}-\frac{\beta^{(\ell)}}{2\alpha^{(\ell)}}\,. \tag{126}$$

While using the approximated gradient $\mathbf{q}(\mathbf{z})$ in our numerical experiments (*c.f.* Section 6), the property (125) always holds for $\ell > 0$.

In our numerical experiments we could sometimes observe increasing energies $f(\tilde{\mathbf{z}}^{(\ell)})+g(\tilde{\mathbf{z}}^{(\ell)})$, but they were always accompanied by decreasing distances $\Delta^{(\ell)}$, leading to a convergent state. In these cases the energies in the convergent state are always lower than the initial energy $f(\tilde{\mathbf{z}}^{(0)})+g(\tilde{\mathbf{z}}^{(0)})$.

When using the exact gradient ∇f, we did not observe increasing energy values in our experiments. However, since the computation of $\mathbf{q}$ is a lot faster and we could achieve good results with the approximated gradient, we regard $\mathbf{q}$ as a more efficient approximation of ∇f.

For a constant step size $\beta^{(\ell)} = \beta$, the sequence $\{\delta^{(\ell)}\}_{l=0}^{\infty}$ may not be monotonically decreasing, so that the convergence theory provided in [25] may not be applicable. This can be fixed be employing a variable $\beta^{(\ell)}$ in Algorithm 1, following the proof of Lemma 4.6 in [25].

Thus for every ℓ we compute the auxiliary variable $\nu := (\delta_{\ell-1}+\frac{L^{(\ell)}}{2})/(c+\frac{L^{(\ell)}}{2})$ and set

$$\beta^{(\ell)} = \frac{\nu-1}{\nu-\frac{1}{2}+c}\,. \tag{127}$$

As initialisation we set $\delta^{(-1)}$=1.

6 Numerical evaluation

In this section we discuss some numerical experiments as well as important observations. A demo with the experiment discussed in Fig. 1 is available online[1].

In all the experiments with our method, the stopping criterion was set to a test on the relative change in the objective function ($< 10^{-8}$), evaluated on $\tilde{\mathbf{z}}^{(\ell)}$ in the inner iPiano loop and on $\mathbf{z}^{(k)}$ in the outer loop. Note that in the outer loop different albedos are used, *i.e.* the energies $f(\mathbf{z}^{(k)},\rho^{(k)})+g(\mathbf{z}^{(k)})$ and $f(\mathbf{z}^{(k+1)},\rho^{(k+1)})+g(\mathbf{z}^{(k+1)})$ are being evaluated. Also the maximum number of iterations was set to 100 in the inner iPiano loop and 500 in the outer loop, if not specified otherwise.

[1] `https://github.com/yqueau/optimized_ps`

6.1 Computational aspects of iPiano

Let us first recall that for a function $f : \mathbb{R}^n \to \mathbb{R}$ with a Lipschitz continuous gradient, such that

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|_2 \leq L^{\nabla f} \|\mathbf{x} - \mathbf{y}\|_2 \tag{128}$$

for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ we have

$$|f(\mathbf{x}) - f(\mathbf{y}) - \langle \nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle| \leq \frac{L^{\nabla f}}{2} \|\mathbf{x} - \mathbf{y}\|_2^2 \tag{129}$$

for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, see *e.g.* [27] Theorem in 3.2.12. This leads to the property

$$f(\mathbf{x}) \leq f(\mathbf{y}) + \langle \nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle + \frac{L^{\nabla f}}{2} \|\mathbf{x} - \mathbf{y}\|_2^2 \tag{130}$$

for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, which is also the subject of the descent lemma, *c.f.* [25] Lemma 4.1. In the iPiano algorithm only the necessary condition (130) is tested with $\mathbf{x} = \tilde{\mathbf{z}}^{(\ell+1)}$ and $\mathbf{y} = \tilde{\mathbf{z}}^{(\ell)}$ and used to derive a local Lipschitz constant. By this, one can allow step sizes leading to a steeper (better) descent in f. In our experiments we often encountered rather low local Lipschitz constants, some examples at hand of the *Cat* data set ([36], see also Figure 5) are depicted in Figure 2. The Lipschitz constants are at first monotonically decreasing, since in each iteration the Lipschitz constant of the previous iteration divided by 1.05 is the first guess for testing (130). Sometimes we encountered increasing local Lipschitz constants towards the end of an iPiano instance. These would then lead to decreasing step sizes $\alpha^{(\ell)}$, such that finally the break criteria for the iPiano algorithm would be fulfilled. An example is depicted in Figure 3 (a).

While in most iterates in our experiments the energy $f(\mathbf{z}) + g(\mathbf{z})$ was decreasing, sometimes it was slightly increasing towards the end of the sequence of iPiano iterations, see Figure 3 (b). We conjecture that this is related to approximated gradients $\mathbf{q}$, which do not deliver a descent direction with $\langle \mathbf{q}, \nabla f \rangle \geq 0$, see also Figure 4.

We did not observe any spikes in the sequence of local Lipschitz or increasing energies when the exact gradient (43) was used. Therefore the use of the exact gradient would lead to a somehow smoother and faster convergence in terms of number of iterations. However, $\mathbf{q}(\mathbf{z})$ can be computed much faster than $\nabla f(\mathbf{z})$ and we did not observe the exact gradient leading to local minima of (19) with significantly smaller energies, so that we still regard the use of the approximated gradient as the more feasible alternative. In detail, the average computation time (100 evaluations) of the exact gradient is roughly 55 seconds, whereas the simplified gradient can be evaluated in 0.13 seconds, which results in a speedup factor of more than 400.

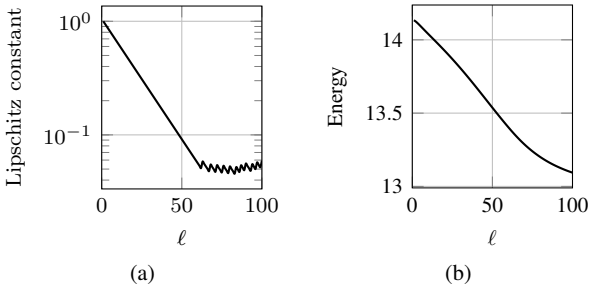

Fig. 2 Account of a typical optimisation cycle in the *Cat* experiment, $k = 1$: (a) local Lipschitz constants obtained by the lazy backtracking strategy, monotonically decreasing until $\ell = 62$; (b) objective function $f + g$ as a function of the iPiano iterations count ℓ. At $\ell = 100$ the albedo is recomputed.

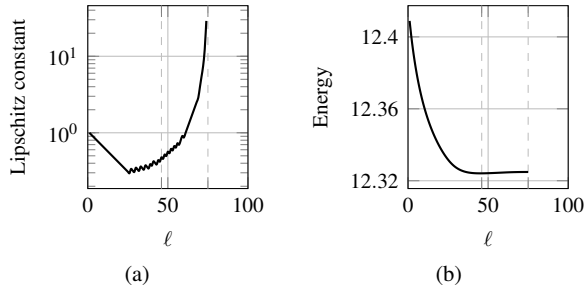

Fig. 3 The *Cat* experiment, $k = 2$: (a) local Lipschitz constants obtained by the lazy-backtracking strategy, monotonically decreasing until $\ell = 26$. At $\ell = 75$ (dashed line) the increase generates a small $\alpha^{(\ell)}$, therefore the iPiano break criteria is fulfilled; (b) objective function $f + g$ as a function of the iPiano iterations count ℓ. Starting at $\ell = 46$ (dashed line) the objective function slightly increases. This is not contrary to the convergence theory, since the descent property is fulfilled for a majorising sequence.

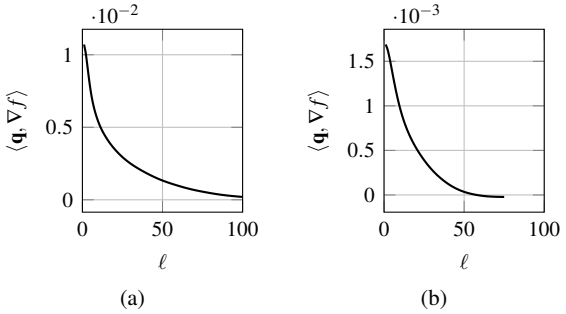

Fig. 4 The *Cat* experiment: $\langle \mathbf{q}, \nabla f \rangle$ for (a) $k = 1$ and (b) $k = 2$. For non-negative values the vector $-\mathbf{q}(\mathbf{z}^{(\ell)})$ is a descent direction.

6.2 Numerical results

Figure 5 presents the test data that we use in this paper. It consists of five real-world scenes captured under 96 different known illuminants $\mathbf{s}^i$, provided in [36]. If not specified otherwise, in our experiments we used $m = 20$ evenly sampled (with indices $1, 6, \ldots, 91, 96$) out of the original 96 RGB images, which we converted to grey levels. Two of the sets present diffuse reflectance (*Cat* and *Pot*), while two other exhibit broad specularities (*Bear* and *Buddha*) and one presents sparse specular spikes (*Ball*). Since the ground truth

normals are also provided in [36], the estimated normals can be computed from the final depth map according to (3), and compared to the exact ground truth. For evaluation, we indicate the mean angular error (MAE) (in degrees) over the reconstruction domain Ω.

Let us consider the *Cat* data set in some detail, as it consists of a diffuse scene that fits rather well our modelling assumptions.

In our first experiment, we test if the optimisation of the reprojection error is equivalent to obtaining better quality. We let our algorithm run for 1000 outer iterations k (approx. 1 hour on a recent Intel Core i7 processor, using non-optimised Matlab code), and study the evolution of two criteria: the reprojection error, whose minimum is sought by our algorithm; and the MAE, which indicates the overall accuracy of the 3D reconstruction, *c.f.* the upper two images within Figure 6. The displayed convergence graphs indicate that each iteration from Algorithm 1 not only decreases the value of the objective function $f+g$ (which is approximately equal to the reprojection error $\mathscr{E}_{\mathscr{R}} = f$), but also the MAE. This confirms our conjecture that finding the best possible explanation of the images yields more accurate 3D reconstructions. In the context of possible model formulations, this means that the reprojection error might be a natural candidate for an objective function.

In the two graphs in the middle in Figure 6 we study the results of our method compared to other PS strategies based on least-squares: the classical PS framework [41] consisting in estimating in a least squares sense the normals and the albedo, and integrating them afterwards, and the recent differential ratios procedure from [21], forcing Lambertian reflectance and least-squares estimation, for fair comparison. This means especially that in the approach from [21] we disable the near-point light setting, which allows us to compare with an approach, that also focuses on the depth instead of normals and uses the classic PS assumptions. The method with this setting is also the subject of [33]. Both other approaches rely on linear least squares: they are thus by far faster than the proposed approach (here, a few seconds, versus a few minutes with ours). Yet, in terms of accuracy, these methods are outperformed by our approach, no matter the noise level or the number of images.

The two graphs at the bottom of Figure 6 display a study about the influence of λ in our model (19), without fixed limits on the number of iterations. For high λ the number of iterations is very low, but the improvement in the MAE over the classic PS prior ($MAE = 8.83$) is very small. When decreasing λ the quality of the reconstructed surface increases, but so does the number of iterations. The experiment was again conducted on the *Cat* data set. Summarising the result, our method behaves reasonably robust versus the choice of λ.

Table 1 Reconstruction errors (MAE, in degrees) obtained for preprocessed input images using the approach from [42], except for the method based on sparse Bayesian learning (SBL) [14]. Each of the listed values is the mean of 5 experiments with randomly chosen images. For fair comparison, the MAEs for classic PS and for SBL are calculated on the final surface, *i.e.* using the normals calculated by finite differences from the final depth map, rather than the (non-integrable) normals estimated in the first step. Regarding the ratio procedure, we applied the code from [21] directly on the grey level data.

	Cat	Pot	Bear	Buddha	Ball
Classic PS [41]	9.43	9.38	8.51	14.86	3.32
Differential ratios [21]	9.00	9.42	8.78	14.89	3.21
SBL [14]	9.14	**9.05**	8.50	**14.06**	**2.98**
Our method (500 iter.)	**8.97**	9.09	**8.42**	14.59	3.14

We also evaluate histograms of the angular error in Figure 7. In comparison with the classic PS approach [41], with our method we have more pixel with low angular error (*i.e.* $<$ 8, the first four bars). Consequently with our method we have less pixel with higher angular errors, this is especially visible at hand of the pixel with angular error in $[8,20)$.

By making the input images Lambertian via low-rank preprocessing [42], we can make a reasonable comparison for the whole test dataset. In order to attempt a comparison of methods arising at robustness versus outliers we also compare our approach to the robust method proposed in [14], using the reference implementation available online. This method is based on sparse Bayesian learning (SBL) and the underlying model explicitly includes outliers. Therefore for the SBL-based method we did not use preprocessing and the model parameter λ for this method was set to 10^{-6}, as suggested in [14]. Per dataset and method we conducted 5 experiments with 20 randomly chosen images, since this setting allows to take over the considerations of the experiments conducted up to this point. For choosing the images we used the Matlab command `randperm`$(96,20)$, where the seed for the random number generator was set to $1,\ldots,5$. Table 1 shows that our postprocessing method can improve the accuracy of the prior derived with the classical PS approach [41]. Moreover our approach outperforms the method based on differential ratios [21] and is competitive with the robust SBL-based method [14]. If one is interested only in the surface normals, further results with methods using the full sets of 96 images can be found at `https://sites.google.com/site/photometricstereodata/`.

Although the sample size of 25 experiments is relatively small in a statistical context, let us quantify the aforegoing evaluation of Table 1. To that end we document the MAEs for each experiment in Table 2 and state confidence intervals of the change in the MAE, if we compare our approach with another method. Here we only note that for experiments with comparable settings, with 95% certainty we expect the true mean to be within the interval $\mathscr{I}_{0.95}^{\text{Classic}}$, $\mathscr{I}_{0.95}^{\text{Diff}}$

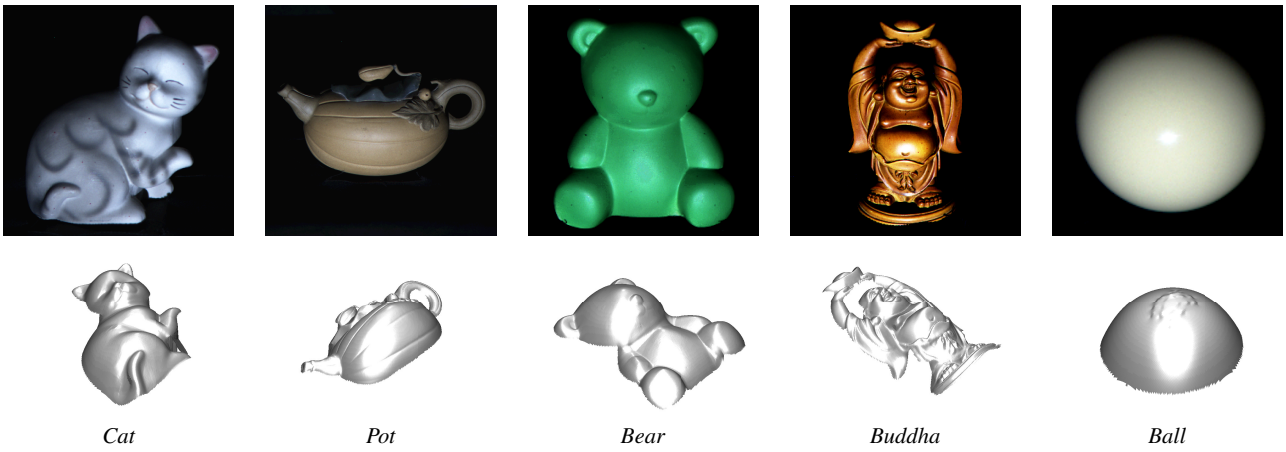

Fig. 5 Test data (brightened and cropped to enhance visualisation) and 3D-reconstructions obtained after 500 iterations k of Algorithm 1.

and $\mathscr{I}_{0.95}^{\mathrm{SBL}}$ respectively, where we measure the difference between the MAE (in degree) from classic PS [41], differential ratios [21] and SBL [14] respectively and the MAE from our method, *i.e.* negative values imply that our approach achieves result with a lower MAE. From the data that is represented in Table 2 we derive

$$\mathscr{I}_{0.95}^{\mathrm{Classic}} = (-0.329, -0.184)\,, \tag{131}$$
$$\mathscr{I}_{0.95}^{\mathrm{Diff}} = (-0.314, -0.123)\,, \tag{132}$$
$$\mathscr{I}_{0.95}^{\mathrm{SBL}} = (-0.279, 0.472)\,. \tag{133}$$

If we the compute confidence intervals with 99.9% certainty, the intervals $\mathscr{I}_{0.999}^{\mathrm{Classic}}$ and $\mathscr{I}_{0.999}^{\mathrm{Diff}}$ again contain only negative values:

$$\mathscr{I}_{0.999}^{\mathrm{Classic}} = (-0.389, -0.125)\,, \tag{134}$$
$$\mathscr{I}_{0.999}^{\mathrm{Diff}} = (-0.391, -0.045)\,, \tag{135}$$
$$\mathscr{I}_{0.999}^{\mathrm{SBL}} = (-0.584, 0.777)\,. \tag{136}$$

Therefore we can expect, that our approach outperforms its immediate competitors classic PS [41] and differential ratios [21].

After computing the confidence interval

$$\mathscr{I}_{0.4}^{\mathrm{SBL}} = (0, 0.193)\,, \tag{137}$$

we see that with only 40% certainty we can say that our approach performs worse than SBL [14]. At this point we want to emphasize, that in contrast to [14] our underlying model (19) is not robust with respect to outliers. Despite this fact the consideration of a nonlinear model makes our approach competitive with the robust method from [14].

Now we return to the images with indices $1, 6, \ldots, 91, 96$. The 3D reconstruction results obtained with the full pipeline are shown in Figure 8. In comparison with Figure 5, artefacts due to specularities are clearly reduced.

Table 2 Reconstruction errors (MAE, in degrees) obtained for randomly chosen sets of images from each dataset. The first column contains the method and the seed used by the random number generator. The mean values are contained in Table 1.

Classic PS [41]	Cat	Pot	Bear	Buddha	Ball
seed 1	9.69	9.54	7.96	14.94	3.14
seed 2	9.26	10.02	8.25	15.13	3.55
seed 3	9.07	9.15	10.77	14.41	3.34
seed 4	9.94	9.06	8.12	15.11	3.46
seed 5	9.20	9.11	7.44	14.73	3.11
Diff. ratios [21]	**Cat**	**Pot**	**Bear**	**Buddha**	**Ball**
seed 1	8.98	9.38	8.22	14.87	2.97
seed 2	8.51	9.78	8.70	15.29	3.42
seed 3	8.80	9.52	11.48	14.66	3.40
seed 4	9.91	9.23	8.12	14.97	3.15
seed 5	8.77	9.21	7.39	14.66	3.10
SBL [14]	**Cat**	**Pot**	**Bear**	**Buddha**	**Ball**
seed 1	9.45	9.52	10.97	14.87	3.34
seed 2	9.17	9.78	8.71	14.48	3.36
seed 3	8.71	8.60	8.73	13.08	2.67
seed 4	9.61	8.70	7.32	14.39	3.02
seed 5	8.74	8.66	6.79	13.46	2.53
Our method	**Cat**	**Pot**	**Bear**	**Buddha**	**Ball**
seed 1	9.21	9.11	7.81	14.55	2.96
seed 2	8.93	9.65	8.11	14.90	3.43
seed 3	8.63	9.05	10.90	14.36	3.29
seed 4	9.58	8.80	7.91	14.71	3.09
seed 5	8.50	8.83	7.36	14.45	2.95

In addition we visualize the reprojection error with respect to preprocessed [42] data in Figure 9 for certain datasets. At first glance, visualizing errors below $2.5 \cdot 10^{-3}$ seems not very meaningful. However the data provided in [36] consists of 16-bit images. Therefore even for grey images the lowest positive change of grey levels is approximately $1.5 \cdot 10^{-5}$. The reprojection error is clearly reduced with the pro-

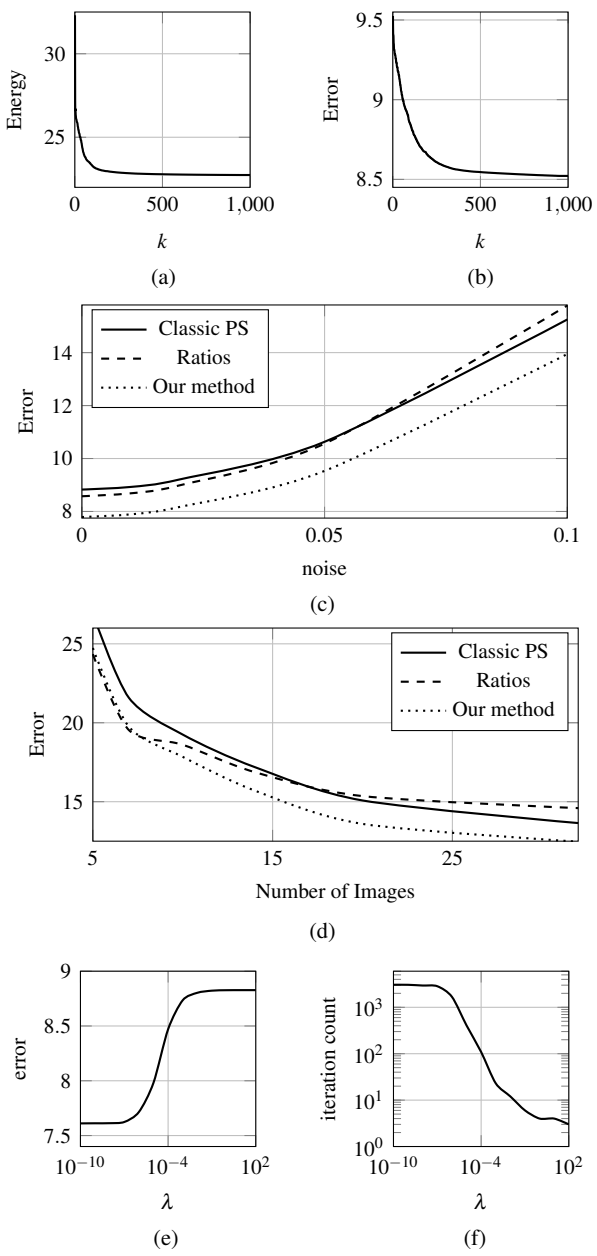

Fig. 6 The *Cat* experiment: (a) objective function $f+g$ as a function of the iterations count k; (b) MAE between the reconstructed surface and the ground truth; (c) MAE for competing methods for increasing noise levels (we indicate the standard deviation of the additive, zero-mean Gaussian noise, as a percentage of the maximum intensity); (d) ditto for increasing numbers of input images, with 0.1 noise level; (e) MAE, in degrees, depending on the choice of λ; (f) number of iterations until a break criteria is fulfilled depending on the choice of λ

posed method, especially in regions with relatively difficult geometry, *e.g.* where edges occur or structures that may result in shadows. This shows that our method may have clear advantages when dealing with complex geometries, which is not perceivable when considering just the measured error improvement documented in Table 1.

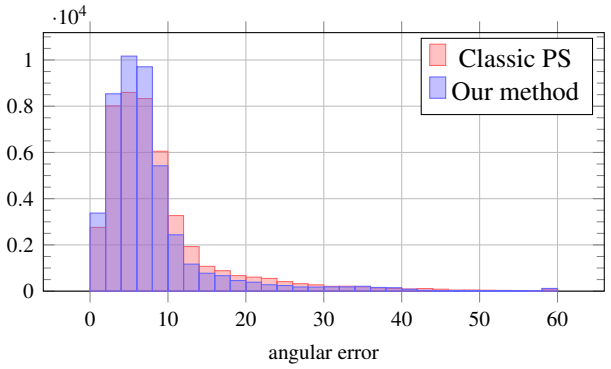

Fig. 7 Histograms of angular errors (in degree) for the *Cat* experiment. Depicted are the number of pixel with angular error in $[0,2)$, $[2,4)$ …, $[56,58)$, $[58,180]$ for classic PS [41] and the proposed method.

As mentioned before, the *Cat* data set fits our modelling assumptions rather well, as it consists of a diffuse scene. However, shadows still pose a problem in some parts of this data set. If we use the groundtruth normals and the images without preprocessing we observe a mean reprojection error (MRE) of $4.79 \cdot 10^{-5}$. With our method we obtain a MRE of $2.30 \cdot 10^{-5}$, which is below the error of the groundtruth. The albedo map for the groundtruth normals is chosen optimal with respect to a quadratic penaliser. Consequently the albedo is influenced by outliers like shadows, resulting in that higher MRE for the groundtruth.

The question remains if the reduction of the reprojection error below the level of the groundtruth normals can lead to a lower MAE. In this context it should be noted that a hypothetical algorithm, which generates a depth with the same reprojection error as the groundtruth normals, will in general not result in a MAE of zero. As long as the groundtruth normals are non-integrable, a PS method generating the depth will not reach a MAE of zero, except through approximation errors in numerical differentiation in some pathological cases. This issue also raises the need for an error measure in terms of depth, used in conjunction with a groundtruth depth.

To further understand the relation between MAE and reprojection error we also conduct an experiment with synthetic data, where we use a part of a sphere with 20 different lighting vectors. The spherical sector is choosen such that there are no shadows. Since this setting fits our modelling assumptions the reprojection error of the groundtruth normals is zero. When computing the normals from the groundtruth depth through finite differences, reprojection error and MAE are substantially higher than zero, as is documented in Table 3. Compared to the groundtruth depth, classic PS [41] leads to a MRE that is decreased by 93% and a MAE that is decreased by 90%. Compared to the result of classic PS, which is the prior for our postprocessing, our method still leads to a 46% decrease in MRE, but the MAE is only de-

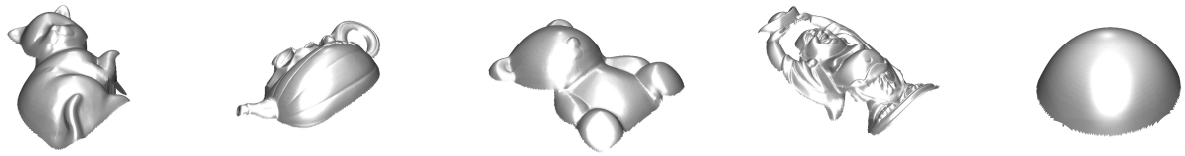

Fig. 8 3D-reconstruction results using the *full pipeline*, consisting of a preprocessing [42], followed by classic PS [41], and finally the proposed method.

Fig. 9 Reprojection error, using white for $2.5 \cdot 10^{-3}$ and black for zero. Displayed are the results via (left) classic PS [41] and (right) the proposed method.

creased by 2.5%. This relatively small quality gain may stem from the fact that classic PS [41] already leads to a very good reconstruction for a regular object like a spherical sector. However, as in previous experiments, the decrease of the reprojection error again leads to a smaller MAE, if only by a small percentage. Our synthetic experiment shows that under ideal model assumptions the MRE and MAE appear to be correlated error measures. However, in our impression there is still not an ideal error measure for PS, which should include also errors in albedo computation.

Table 3 Synthetic sphere data: MRE and MAE (in degrees) for different depth maps. For the groundtruth depth the actual coordinates are used, and not the groundtruth normals. For classic PS as before, the final depth map is used rather than the non-integrable normals produced in the first step. With our method, while the MRE is strongly reduced, there is only a small quality gain in terms of MAE when comparing to classic PS. Nevertheless, the employed error measures appear to correlate in the experiment.

	MRE	MAE
Groundtruth depth	$4.080 \cdot 10^{-6}$	0.44807
Classic PS [41]	$3.013 \cdot 10^{-7}$	0.04325
Our method	$1.634 \cdot 10^{-7}$	0.04216

7 Conclusion

We have shown the benefits of recent, high performing numerical methods in the context of photometric stereo. Let us emphasise that only by considering such recent developments in numerical optimisation methods complex models as arising in PS can be handled with success. Our results show that a reasonable quality gain can be achieved in this way while at the same time the mathematical proceeding can be validated rigorously.

Our experimental investigation has shown what can be expected from the basic iPiano method as well as by computational simplifications as proposed by us in terms of an approximated gradient. In particular we have shown that it may not be easy to interprete relevant properties of computed iterates.

A more detailed view on the computational results reveals that remaining inaccuracies seem to be mostly due to shadows and highlights, edges and depth discontinuities. Thus, an interesting perspective of our work would be to use more robust estimators, which would ensure both robustness to outliers [14, 34] and improved preservation of edges [6].

References

1. Bähr, M., Breuß, M., Quéau, Y., Boroujerdi, A.S., Durou, J.D.: Fast and accurate surface normal integration on non-rectangular domains. Computational Visual Media **3**, 107–129 (2017)
2. Bartal, O., Ofir, N., Lipman, Y., Basri, R.: Photometric stereo by hemispherical metric embedding. Journal of Mathematical Imaging and Vision (to appear). URL `https://doi.org/10.1007/s10851-017-0748-y`

3. Basri, R., Jacobs, D., Kemelmacher, I.: Photometric stereo with general, unknown lighting. International Journal of Computer Vision **72**, 239–257 (2007)
4. Chabrowski, J., Kewei, Z.: On variational approach to photometric stereo. Annales de l'Institut Henri Poincaré (C) Analyse non linéaire **10**(4), 363–375 (1993)
5. Clark, J.J.: Active photometric stereo. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 29–34 (1992)
6. Durou, J.D., Aujol, J.F., Courteille, F.: Integrating the normal field of a surface in the presence of discontinuities. In: Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR), *Lecture Notes in Computer Science*, vol. 5681, pp. 261–273. Springer (2009)
7. Gotardo, P.F.U., Simon, T., Sheikh, Y., Matthews, I.: Photogeometric scene flow for high-detail dynamic 3D reconstruction. In: Proc. IEEE Int. Conf. Computer Vision (ICCV), pp. 846–854 (2015)
8. Harker, M., O'Leary, P.: Regularized reconstruction of a surface from its measured gradient field. Journal of Mathematical Imaging and Vision **51**(1), 46–70 (2015)
9. Hinkley, D.V.: On the ratio of two correlated normal random variables. Biometrika **56**(3), 635–639 (1969)
10. Hoeltgen, L., Quéau, Y., Breuß, M., Radow, G.: Optimised photometric stereo via non-convex variational minimisation. In: British Machine Vision Conference (BMVC) (2016). URL `https://doi.org/10.5244/C.30.36`
11. Horn, B.K.P.: Robot Vision. The MIT Press (1986)
12. Horn, B.K.P., Woodham, R.J., Silver, W.M.: Determining shape and reflectance using multiple images. Technical Report MIT AITR-490, MIT (1978)
13. Horn, R.A., Johnson, C.R.: Topics in Matrix Analysis. Cambridge University Press (1994)
14. Ikehata, S., Wipf, D., Matsushita, Y., Aizawa, K.: Photometric stereo using sparse Bayesian regression for general diffuse surfaces. IEEE Transactions on Pattern Analysis and Machine Intelligence **36**(9), 1816–1831 (2014)
15. Ju, Y.C., Tozza, S., Breuß, M., Bruhn, A., Kleefeld, A.: Generalised perspective shape from shading with Oren-Nayar reflectance. In: British Machine Vision Conference (2013). URL `http://doi.org/10.5244/C.27.42`
16. Khanian, M., Boroujerdi, A.S., Breuß, M.: Photometric stereo for strong specular highlights. Computational Visual Media (to appear). URL `https://arxiv.org/abs/1709.01357`
17. Kozera, R.: Existence and uniqueness in photometric stereo. Applied Mathematics and Computation **44**, 1–103 (1991)
18. Lambert, J.H.: Photometria. Klett, Augsburg (1760)
19. Magnus, J.R., Neudecker, H.: Matrix differential calculus with applications to simple, Hadamard, and Kronecker products. Journal of Mathematical Psychology **29**, 474–492 (1985)
20. Magnus, J.R., Neudecker, H.: Matrix Differential Calculus with Applications in Statistics and Econometrics, 3rd edn. John Wiley & Sons (2007)
21. Mecca, R., Quéau, Y., Logothetis, F., Cipolla, R.: A single-lobe photometric stereo approach for heterogeneous material. SIAM Journal on Imaging Sciences **9**(4), 1858–1888 (2016)
22. Mecca, R., Rodolà, E., Cremers, D.: Realistic photometric stereo using partial differential irradiance equation ratios. Computers & Graphics **51**, 8–16 (2015)
23. Moreau, J.J.: Proximité et dualité dans un espace Hilbertien. Bulletin de la Société Mathématique de France **93**, 273–299 (1965)
24. Ochs, P.: Unifying abstract inexact convergence theorems for descent methods and block coordinate variable metric iPiano. Tech. rep., Saarland University (2016)
25. Ochs, P., Chen, Y., Brox, T., Pock, T.: iPiano: Inertial proximal algorithm for non-convex optimization. SIAM Journal on Imaging Sciences **7**(2), 1388–1419 (2014)
26. Onn, R., Bruckstein, A.: Integrability disambiguates surface recovery in two-image photometric stereo. International Journal of Computer Vision **5**, 105–113 (1990)
27. Ortega, J.M., Rheinboldt, W.C.: Iterative Solutions of Nonlinear Equations in Several Variables. New York Academic (1970)
28. Papadhimitri, T., Favaro, P.: Uncalibrated near-light photometric stereo. In: British Machine Vision Conference (2014). URL `http://doi.org/10.5244/C.28.128`
29. Petersen, K.B., Pedersen, M.S.: The matrix cookbook (2012). Available from `https://www.math.uwaterloo.ca/~hwolkowi/matrixcookbook.pdf`
30. Pollock, D.S.G.: Tensor products and matrix differential calculus. Linear Algebra and Its Applications **67**, 169–193 (1985)
31. Quéau, Y., Durix, B., Wu, T., Cremers, D., Lauze, F., Durou, J.D.: LED-based photometric stereo: Modeling, calibration and numerical solution. Journal of Mathematical Imaging and Vision (to appear). URL `https://dx.doi.org/10.1007/s10851-017-0761-1`
32. Quéau, Y., Lauze, F., Durou, J.D.: A L^1-TV algorithm for robust perspective photometric stereo with spatially-varying lightings. In: Scale Space and Variational Methods in Computer Vision (SSVM), *Lecture Notes in Computer Science*, vol. 9087, pp. 498–510 (2015)
33. Quéau, Y., Mecca, R., Durou, J.D.: Unbiased photometric stereo for colored surfaces: A variational approach. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4359–4368 (2016)
34. Quéau, Y., Wu, T., Lauze, F., Durou, J.D., Cremers, D.: A non-convex variational approach to photometric stereo under inaccurate lighting. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 99–108 (2017)
35. Reddy, D., Agrawal, A., Chellappa, R.: Enforcing integrability by error correction using l_1-minimization. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2350–2357 (2009)
36. Shi, B., Mo, Z., Wu, Z., Duan, D., Yeung, S.K., Tan, P.: A benchmark dataset and evaluation for non-Lambertian and uncalibrated photometric stereo. IEEE Transactions on Pattern Analysis and Machine Intelligence (to appear). URL `https://doi.org/10.1109/tpami.2018.2799222`
37. Smith, W., Fang, F.: Height from photometric ratio with model-based light source selection. Computer Vision and Image Understanding **145**, 128–138 (2016)
38. Tozza, S., Mecca, R., Duocastella, M., Del Bue, A.: Direct differential photometric stereo shape recovery of diffuse and specular surfaces. Journal of Mathematical Imaging and Vision **56**(1), 57–76 (2016)
39. Wöhler, C.: 3D Computer Vision. Springer-Verlag (2013)
40. Woodham, R.J.: Photometric stereo: A reflectance map technique for determining surface orientation from a single view. In: Proceedings of the 22nd SPIE Annual Technical Symposium, *Proceedings of the International Society for Optical Engineering*, vol. 155, pp. 136–143 (1978)
41. Woodham, R.J.: Photometric method for determining surface orientation from multiple images. Optical Engineering **19**(1), 134–144 (1980)
42. Wu, L., Ganesh, A., Shi, B., Matsushita, Y., Wang, Y., Ma, Y.: Robust photometric stereo via low-rank matrix completion and recovery. In: Asian Conference on Computer Vision (ACCV), *Lecture Notes in Computer Science*, vol. 6494, pp. 703–717. Springer Berlin Heidelberg (2010)
43. Zeisl, B., Zach, C., Pollefeys, M.: Variational regularization and fusion of surface normal maps. In: IEEE International Conference on 3D Vision (3DV), pp. 601–608 (2014)