Photometric Bundle Adjustment for Dense Multi-View 3D Modeling

Amaël Delaunoy, Marc Pollefeys

Photometric Bundle Adjustment for Dense Multi-View 3D Modeling

Amaël Delaunoy
ETH Zürich
Amael.Delaunoy@inf.ethz.ch

Marc Pollefeys
ETH Zürich
Marc.Pollefeys@inf.ethz.ch

Abstract

Motivated by a Bayesian vision of the 3D multi-view reconstruction from images problem, we propose a dense 3D reconstruction technique that jointly refines the shape and the camera parameters of a scene by minimizing the photometric *reprojection error between a generated model and the observed images, hence considering all pixels in the original images. The minimization is performed using a gradient descent scheme coherent with the shape representation (here a triangular mesh), where we derive evolution equations in order to optimize both the shape and the camera parameters. This can be used at a last refinement step in 3D reconstruction pipelines and helps improving the 3D reconstruction's quality by estimating the 3D shape and camera calibration more accurately. Examples are shown for multi-view stereo where the texture is also jointly optimized and improved, but could be used for any generative approaches dealing with multi-view reconstruction settings (i.e. depth map fusion, multi-view photometric stereo).*

1. Introduction

Reconstructing the 3D shape from multiple images has been one of the main challenges in computer vision and has been widely studied. A Bayesian way of addressing the multi-view reconstruction problem is to see it as the *inverse problem* of the image formation process. This process of image generation implies being able to derive a model of such a scene, denoted by $\mathbf{\Omega}$. This typically contains the scene geometry (i.e. the surface $\mathcal{S}$), a camera model (i.e. the pinhole camera model Π), the surface properties (i.e. the reflectance), the lighting conditions, etc. If, given those parameters of $\mathbf{\Omega}$, we are able to generate an image $\overline{R}$, we can compare it to the observed images $\mathbf{I} = \{I_1, I_2, \dots, I_m\}$. The best scene $\hat{\mathbf{\Omega}}$ can be found by maximizing the joint probability of a scene given the images:

$$\hat{\mathbf{\Omega}} = \arg\max_{\mathbf{\Omega}} \{p(\mathbf{\Omega}|\mathbf{I})\} = \arg\max_{\mathbf{\Omega}} \{p(\mathbf{I}|\mathbf{\Omega})\, p(\mathbf{\Omega})\}\,. \quad (1)$$

The terms $p(\mathbf{\Omega})$is the prior term on the scene (which may typically correspond to surface smoothing criteria, constrains on the surface texture or the camera parameters, or simply an initial guess for the model). $p(\mathbf{I}|\mathbf{\Omega})$ corresponds to the likelihood of a generated image for a given shape, appearance and cameras. It measures the similarity between the generated images and the observed images.

A simplification of the generative model is to consider as observation previously detected 2D features in images, along with their corresponding matches in other images. Finding the camera parameters from such information is known as structure-from-motion [4, 15]. Such a model could be estimated by finding the calibration Π and a set of 3D points $\mathbf{x}$ whose projections in the images are as close as possible from the original observations. If we consider a Gaussian noise model in the observation, maximizing this likelihood naturally leads to minimizing the geometric error between the projection of the 3D points and their corresponding 2D measurements (e.g. , 2D feature positions) to refine both the camera parameters and a sparse reconstruction in a single framework. This is known as *Geometric* Bundle Adjustment (BA) [21, 13], and has been successfully applied to various sparse 3D reconstruction scenari, mostly minimizing a *Geometric* Reprojection Error:

$$E(\mathbf{x}, \Pi) = \sum_i f_i(\mathbf{x}, \Pi)^2\,,$$

where f_i is the geometric error between observation i and the projection of the 3D point into the image.

In contrast, an alternative way is to directly consider the maximum likelihood of the generative model described in Equation (1), by finding a model that best explains the observed images. In this case, the observed data no longer consists of extracted features like in the case of *GBA*, but directly comes from the image measurements. In computer vision, this typically corresponds to intensity values of a color image, but the concept naturally generalizes to any 2-dimensional signals coming from vision sensors. $p(\mathbf{I}|\mathbf{\Omega})$ is typically derived from an image noise model, and is often represented as a Normal (or Gaussian) distribution function, e.g. $p(\mathbf{I}|\mathbf{\Omega}) \propto \prod_i \prod_{\mathbf{p}} e^{-(I_i(\mathbf{p}) - \overline{R}_{\mathcal{S},\Pi_i}(\mathbf{p}))^2}$, where

$\overline{R}_{\mathcal{S},\Pi_i}(\mathbf{p})$ is the intensity pixel value induced by the generative model for image i. The reconstruction problem can naturally be formulated as minimizing the following "*photometric*" energy functional [2, 3, 7, 19]:

$$E(\mathbf{\Omega}) = \sum_i \int_{\mathcal{I}_i} \frac{1}{2} \big(I_i(\mathbf{p}) - \overline{R}_{\mathbf{\Omega},i}(\mathbf{p}) \big)^2 \, \mathrm{d}\mathbf{p} \,, \qquad (2)$$

where $\mathrm{d}\mathbf{p}$ is the area measure on the image i. In the rest of the paper, we omit the dependency on i since this is just a sum over all available images. Note that, in contrast with *GBA*, the error measure between the predicted pixel values and the observed ones is carried out over all pixels of all input images. Instead of the geometric information only (i.e. extracted image feature positions), this paper aims at accounting for the *photometric* information, referred to as *photometric* bundle adjustment (*PBA*).

1.1. Related Work

In recent decades, dense geometry recovery has lead to a large number of efficient techniques in order to obtain dense and accurate 3D models, e.g. see [18, 20] for a comparison of recent approaches in the context of multi-view stereo. While some algorithms are based on dense features or patches [5] others are based on energy minimization techniques. Among those techniques, variational methods have become popular. They differ from the kind of energy they minimize, the way they minimize it or the surface representation they choose. For example [17] uses the Level Set framework using a global image score, [10] uses a convex formulation minimizing a photometric error defined over a discretized grid. In [2] and [23], a mesh refinement technique is proposed, minimizing a photometric cost measure. While all those methods return good results in recovering the 3D shape, only a few of them address the problem of camera calibration from dense data. In the following, we describe related work regarding efforts in joint calibration/geometry estimation focusing on the resolution of reprojection error functionals, i.e. Equation (2).

Calibration and Dense Geometry Estimation
It is well established that 3D reconstruction and camera estimation are tightly linked together, bundle adjustment problems being a good example of how calibration can be improved by jointly estimating the 3D structure and the camera parameters. Surprisingly, until recently, dense surface reconstruction was only considered as a next and/or independent step from the calibration problem. It would be more elegant if one could directly minimize the photometric reprojection error to estimate both shape and camera parameters (and eventually the scene radiance) at the same time.

Georgel et al. [8] propose a unified framework to combine both the geometric and photometric information. As both terms are not homogeneous, it is not clear how to combine and weight them efficiently. In this work, we propose to use the photometric information only, assuming an initial calibration is already provided. It is also worth mentioning the work of [5], which estimates 3D oriented patches, and then minimizes the reprojection error to refine both patches and camera parameters. They show substantial improvements in accuracy for 3D reconstruction, hence showing a photometric-based refinement of the calibration is necessary for high quality multi-view stereo. Both [8] and [5] assume the surface can be represented by planar local patches. Here, we represent the surface as an arbitrary triangular mesh. Real-time structure-from-motion is also possible by using dense tracking and mapping [14]. In [14], the authors use a dense photometric cost to refine the camera poses. Our model extends naturally to intrinsic calibration.

Recently, several authors have been interested in addressing the problem of improving both the calibration and the dense reconstruction in the context of minimizing an energy functional of the type (2). The work in this paper is closely related to the ones described in [22, 25]. In these papers, the authors propose to refine the calibration in the context of multi-view modeling using a variational approach. In [22], only the calibration is optimized, and the equations are derived in the context of uniformly colored shapes. Therefore, it is not possible to refine the camera parameters if a segmentation of the object or the visual hull is not available and does not fall into "binary" images. Similarly, [25] also only works for uniformly colored objects. Instead, we propose a generalization of [22] and [25] to deal with textured and more complex objects.

Aubry et al. [1, 9] proposes a different approach. Without solving the problem in a direct way, they relax the problem between correspondence estimation and camera calibration. They decouple the minimization by first estimating the optical flow between a generated image and the observed one, and then refine the camera poses. The process is iterated using a fixed geometry. While this alternate process allows faster convergence and reduce local minima, unlike [1, 9], we directly solve for both the calibration and the 3D reconstruction in a single framework by directly minimizing the reprojection error. While the underlying energy functional is similar, the proposed optimization is fundamentally different as the parameters (Mesh, Cameras and Texture) are optimized jointly in a single framework, and do not use separate independent steps.

Visibility
One of 3D reconstruction's (and more generally computer vision) most challenging problems is the visibility information. While most techniques deal with visibility more or less explicitly (usually as fixed function updates between iterations), very few of them consider the visibility variation

in their formulation. Some consider additional terms such as ballooning terms or silhouette constraints. However, the correct minimization of Equation (2) already contain terms that avoid the empty set to be the optimal solution.

Yezzi and Soatto [24] use the concept of oriented visibility, which implicitly constrains the minimization to make it consistent with the image silhouettes. However their work is limited to convex shapes, and while there is no need for additional constraints during the evolution of the surface, it does not handle self occlusions. Gargallo et al. [7, 6] generalized this idea to non-convex surfaces in the framework of Level-Sets. Delaunoy and Prados [2] extend this concept to discrete polyhedral surfaces, allowing to constrain mesh-based evolution. However, none of the above techniques deal with camera calibration and focus on 3D reconstruction only. In this work, we build on [2] to account for visibility when cameras parameters are optimized as well.

The visibility issue while refining the calibration is partially solved in [22]. However, similarly as in [24], the additional constraint only accounts for silhouette points, and is only valid for convex shapes. We extend this work to arbitrary shapes and derive a similar strategy as in [2].

1.2. Contribution

In this paper we focus on the last stage of the 3D modeling pipeline, i.e. the dense reconstruction using a similar model as [2, 9]. We propose to jointly refine the dense geometry and the camera parameters using the *photometric error*. This error is simply the reprojection error between images of a generative model and the observed images which is directly minimized in the image domain (Equation 2). In order to achieve that goal, we derive equations of the gradient of the energy functional we minimize, accounting for visibility changes [2, 7, 24]. The shape is represented as a triangular mesh, allowing an easier handling of the texture which is also jointly optimized.

In this work, we propose a direct pixel-based bundle adjustment minimization of the photometric reprojection error in order to jointly optimize the full and dense 3D shape as well as the camera parameters and the scene radiance (the texture) by exploiting an image-based reprojection error.

2. Full BA: Problem Formulation

We propose to refine the scene $\mathbf{\Omega}(\mathcal{S}, \Pi)$ from some initial scene $\mathbf{\Omega}^0$, parametrized by its surface $\mathcal{S}$ and calibration Π. While this section describes a variational formulation that is valid for general generative scene models (including depth map integration, shape-from-shading, photometric stereo, etc), Section 3 focuses on the case of a Lambertian scene reconstructed with a generative model including texture, shape and camera parameters.

2.1. Functionals Defined on Visible Surfaces

Equation (2) is minimized on the whole image. However, in order to generate $\overline{R}_{\mathbf{\Omega},i}$, one needs to consider the background B, in order to explain parts of the images where the surface of interest to be modeled does not project into the images. Equation (2) becomes (see [2]):

$$\begin{aligned} E(\mathcal{S},\Pi) &= \int_{\Pi(\mathcal{S})} \big(I - \overline{R}_{\mathcal{S},\Pi}\big)^2 + \int_{\mathcal{I}_i - \Pi(\mathcal{S})} \big(I - B\big)^2 \\ &= \int_{\Pi(\mathcal{S})} \Big[\big(I - \overline{R}_{\mathcal{S},\Pi}\big)^2 - \big(I - B\big)^2\Big] + \int_{\mathcal{I}} \big(I - B\big)^2 . \end{aligned} \tag{3}$$

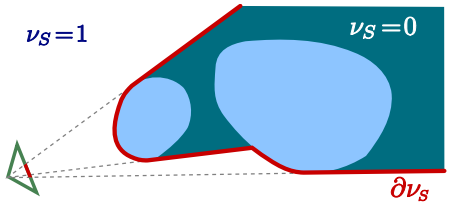

Figure 1. 3D surface $\mathcal{S}$ seen from a camera showing visible and occluded volumes. Only the visibility interface $\partial\nu_{\mathcal{S}}$ can be explained the images.

Minimizing the data fidelity term of Equation (3) is rather difficult and similarly to previous works we use a gradient descent strategy [7, 19, 24]. This is due to the fact that the generative model (mostly the projection and occlusions created by the surface) implicitly accounts for visibility. It is then important to know how the updates on the scene parameters affect the changes in visibility. This function is illustrated in Figure 1. Let $\nu_{\mathcal{S},\Pi}(\mathbf{x})$ be the visibility function $\nu_{\mathcal{S},\Pi} : \mathbb{R}^3 \mapsto [0,1]$ such that:

$$\nu_{\mathcal{S},\Pi_i}(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \text{ is visible from the camera } i, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

In order to minimize Equation (3), let us first rewrite the equation over the (visible) surface. Note that the second term is constant and can be left out in the minimization. Denoting the part in brackets by $f_{\mathbf{\Omega}}$, this gives, by a simple change of variables [19, 24]:

$$E(\mathcal{S},\Pi) = \int_{\mathcal{S}\cup\overline{B}} f_{\mathbf{\Omega}}(\mathbf{x},\mathbf{n}(\mathbf{x}))\ \boldsymbol{\alpha}\cdot\mathbf{n}\ \nu_{\mathcal{S}}(\mathbf{x})\,\mathrm{d}s\,, \tag{5}$$

where we have $\mathrm{d}\mathbf{p} = f_x f_y \dfrac{\mathbf{d}\cdot\mathbf{n}}{\mathbf{d}_z^3}\ \nu_{\mathcal{S}}(\mathbf{x})\ \mathrm{d}s$ for a pinhole camera model, e.g. see [19]. f_x and f_y are the focal parameters in x and y respectively, $\mathbf{n}$ is the surface normal at point $\mathbf{x}$ and $\mathbf{d}$ is the vector pointing from the camera center to $\mathbf{x}$. $\overline{B}$ is the background surface (either a previously estimated surface or a plane behind the object of interest). This generalizes to other parametric camera models. In the following we denote $\boldsymbol{\alpha} = f_x f_y \dfrac{\mathbf{d}}{\mathbf{d}_z^3}$.

2.2. Variational Refinement of the Surface

The choice of the surface representation is rather important and conditions the rest of the minimization. We choose a triangular mesh as it was recently proven to give accurate and impressive results [23]. It also has the advantage to move vertices at their correct location as the final gradient flow is allowed to have tangential components in the evolution of the vertices, and to provide a manifold watertight mesh, suitable for further processing or applications.

Similarly as in [2, 23], we follow a *discretize then minimize* strategy by discretizing the energy functional over a polyhedral representation $\mathbf{X}$ of the surface $\mathcal{S}$. Let $\mathbf{X} = \{\mathbf{x}_1 \dots \mathbf{x}_n\}$ be a piecewise planar triangular mesh, $\mathbf{x}_k$ being the k^{th} vertex of $\mathbf{X}$, and let $\mathcal{S}_j$ be the j^{th} triangle of $\mathbf{X}$. The energy functional (2) we finally minimize is:

$$E(\mathbf{X}, \mathbf{\Pi}) = \sum_j A_j \int_{\mathsf{T}} f_{\mathbf{\Omega}}(\mathbf{x}(\mathbf{u}))\, \boldsymbol{\alpha}(\mathbf{u}) \cdot \mathbf{n}_j\, \nu_{\mathcal{S}}(\mathbf{x}(\mathbf{u}))\, \mathrm{d}\mathbf{u}\,, \tag{6}$$

where $\mathbf{n}_j$ is the normal of the triangle $\mathcal{S}_j$ parametrized by $\mathbf{u}$ of surface area A_j and where the sum is over all the triangles of the mesh $\mathbf{X}$. Over each triangle, points are parametrized using barycentric coordinates $\mathbf{u} = (u, v) \in \mathsf{T} = \{(u, v) | u \in [0, 1] \text{ and } v \in [0, 1 - u]\}$. The term $\mathrm{d}\mathbf{u} = 2\, A_j\, \mathrm{d}s$ corresponds to the surface area element on the triangle. In the following, in order to simplify notations, we omit the dependency in $\mathbf{u}$.

In order to compute the gradient of Equation (6), we consider the evolution of the energy under a small evolution of the surface $\mathbf{X}[t] = \mathbf{X}^0 + t\mathbf{V}$, where $\mathbf{V}$ is a vector field defined on all the vertices $\mathbf{x}$ of the mesh $\mathbf{X}$. The directional derivative of $E(\mathbf{X})$, i.e. $\frac{d}{dt} E(\mathcal{S}[t])\Big|_{t=0}$ is used to compute the final gradient of the energy $E(\mathbf{X})$. The mesh evolution equation is given by the following L^2 gradient descent flow:

$$\begin{cases} \mathbf{X}[0] = \mathbf{X}^0\,, \\ \mathbf{X}[t+1] = \mathbf{X}[t] - dt\; M^{-1} \dfrac{\partial E}{\partial \mathbf{X}}(\mathbf{X}[t])\,, \end{cases} \tag{7}$$

where $\mathbf{X}^0$ is an initial mesh and M is the mass matrix containing the area around a particular vertex. This means that the velocity of one particular vertex depends on the integrated cost on neighboring facets, hence allowing consistent vertex displacements. This gradient descent scheme contains typically two elements: one corresponding to occluding contours, and one for the vertices that do not make strong changes in the visibility. This second part of the gradient typically describes the gradient of vertices on the occluding contours (called the *horizon* term). The way visibility changes at occluding contours is illustrated in Figure 2 and we follow the gradient computations detailed in [2].

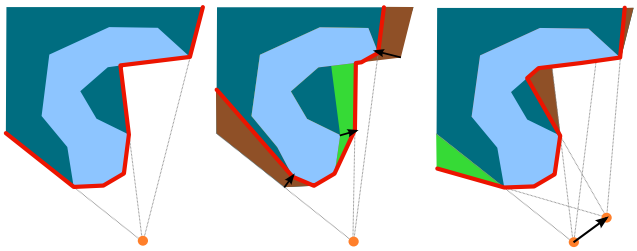

Figure 2. Left: Original discrete mesh and its visibility interface; Middle: Change of the visibility interface when moving a vertex of the mesh $\mathbf{X}$ [2, 7]; Right: Influence of camera center update on the horizon. Moving points on occluding contours or moving the camera center drastically changes the visibility function.

2.3. Camera Refinement

We now consider the same energy functional as a function of the camera parameters Π. We consider a standard pinhole camera model, and parametrize π using a set of parameters (g_i), accounting for the intrinsics (focal, skew parameter and principal point) and extrinsics of the camera.

The extrinsics are the rotation $R(\omega)$ and translation in $\mathbb{SE}(3)$ that are parametrized using an angle-axis representation for the rotation ω, and the optical center of the camera C. It is worth to mention that among the calibration parameters, only the camera center induces changes in the visibility function $\nu_{\mathcal{S},\Pi}$. Figure 2 gives a geometric intuition of what is happening to the visibility interface during the optimization. For a rotation update, the gradient of the visibility function is zero. The camera updates are computed using the following partial differential equations:

$$\frac{\partial E(\mathbf{X}, \mathbf{\Pi})}{\partial \mathbf{\Pi}} = \sum_j A_j \int_{\mathsf{T}} \frac{\partial}{\partial \mathbf{\Pi}} \Big(f_{\mathbf{\Omega}}(\mathbf{x})\, \boldsymbol{\alpha} \cdot \mathbf{n}_j\, \nu_{\mathcal{S}}(\mathbf{x}) \Big)\, \mathrm{d}\mathbf{u}\,. \tag{8}$$

The resolution of Eq. (8) implies classical derivations and standard chain rules and follows similar strategies as described in [9, 22]. For the visibility gradient on the camera center, we first rewrite Eq. (8) as an energy over the volume. We derive how the energy changes when moving the camera similar as for the horizon term in the mesh optimization [2]. The difference is on the volume parametrization which in this case follows the shape in Fig. 2. In practice, this term has less influence as it is computed over the whole image and most points are not on occluding contours.

While we focus on classical pinhole camera intrinsics and extrinsics, we could add more complete calibration models by adapting $\boldsymbol{\alpha}$ and its associated derivatives.

3. Multi-view Stereo Application

In multi-view stereo, the generative model of a scene $\mathbf{\Omega}$ depends not only on the surface shape and camera parameters, but also on the scene radiance, i.e. the texture of the

surface. Let $\mathbf{T} : \mathbb{R}^2 \mapsto \mathbb{R}^3$ be the texture map to optimize. Let the mesh $\mathbf{X}$ be the mesh representation of the surface $\mathcal{S}(\mathbf{X}, \mathbf{T})$. The associated generative model can be defined as $\overline{R}_{\mathbf{\Omega},i}(\mathbf{p}) = T(\Pi_{i,\mathcal{S}}^{-1}(\mathbf{p}))$, where T is the Lambertian texture of the surface. $\Pi_{\mathcal{S}}^{-1}(\mathbf{p})$ is the back-projection of pixel $\mathbf{p}$ onto the surface if it exists, or onto the background B otherwise. The energy we minimize in this context is very similar to reprojection errors used in other related works [2, 7, 9]:

$$E(\mathbf{X}, \mathbf{T}, \mathbf{\Pi}) = \sum_i \int_{\mathcal{I}_i} \frac{1}{2} \left(I_i(\mathbf{p}) - T(\Pi_{i,\mathbf{X}}^{-1}(\mathbf{p})) \right)^2 \, \mathrm{d}\mathbf{p} \,, \tag{9}$$

3.1. Geometry, Calibration and Texture Recovery

Similarly as in the previous section, we minimize the following energy defined over the surface:

$$E(\mathbf{X}, \mathbf{T}, \mathbf{\Pi}) = \sum_j A_j \int_{\mathsf{T}} \left(I_i(\Pi_i(\mathbf{x}(\mathbf{u}))) - T(\mathbf{u}) \right)^2 \boldsymbol{\alpha} \cdot \mathbf{n}_j \, \nu_{\mathcal{S}}(\mathbf{x}(\mathbf{u})) \, \mathrm{d}\mathbf{u}, \tag{10}$$

which is minimized as previously described. The only difference is that the sampling of the residuals (the numerical integration over the triangles) is performed on the texture space rather than the triangles directly. This allows us to control the sampling and make sure it is coherent with the image resolution (details in the experimental section).

A natural shape prior is to penalize non-smooth surfaces. Instead of minimizing the surface area which introduces bias towards minimal surfaces, one may add a smoothness term to penalize variations on the surface normals. This can be achieved by minimizing the following energy functional:

$$E_{\text{reg}}(\mathbf{X}) = \lambda_S \sum_j A_j \int_{\mathsf{T}} |\mathbf{n}_j - \mathbf{h}_j|^2 \, \mathrm{d}\mathbf{u}, \tag{11}$$

where $\mathbf{h}_j$ is an unit vector. Typically $\mathbf{h}_j$ is the average normal on a local neighborhood around the facet j. λ_S is a smoothing parameter. Similarly as in [23], we weight the data term by the squared ratio between the average image depth and the focal length in order to get the energy homogeneous in squared world units, hence having a smoothness parameter stable across different datasets.

Texture Estimation

As mentioned above, $T : \mathcal{S} \to \mathbb{R}^3$ is the estimated radiance on the surface. An obvious choice for the texture $T(\mathbf{x})$ is the closed form solution of Equation (10):

$$T(\mathbf{x}) = \frac{\sum_i I_i(\Pi_i(\mathbf{x})) \, w_i(\mathbf{x})}{\sum_i w_i(\mathbf{x})} \,. \tag{12}$$

$T(\mathbf{x})$ corresponds to the weighted mean color at point $\mathbf{x}$ of the images where $\mathbf{x}$ is visible. We have $w_i(\mathbf{x}) = \boldsymbol{\alpha} \cdot \mathbf{n}_j \, \nu_{\mathcal{S}}(\mathbf{x}(\mathbf{u}))$. Since we want to minimize $\mathbf{X}, \mathbf{T}, \mathbf{\Pi}$ at the same time, we use the following texture evolution:

$$T_{t+1}(\mathbf{u}) = T_t(\mathbf{u}) + dt \sum_i (I_i(\Pi_i(\mathbf{x}(\mathbf{u}))) - T_t) \, w_i(\mathbf{x}) \,. \tag{13}$$

While one could plug Equation (13) directly in Equation (9) and get rid of the texture, handling the texture separately offers significant advantages. For example, it becomes necessary if one wants to optimize more complete reflectance models (albedo, specular coefficient, etc), or want to add more realistic image formation models.

4. Experiments and Results

The proposed photometric bundle adjustment approach is evaluated on several publicly available datasets. We show improvements not only on the dense 3D geometry, but also in the estimated texture of the surface.

The algorithm has been implemented in C++ and is running on a standard 3GHz Linux machine. We use the GPU (using OpenGL Shading Language) for computing visibility by rendering depth maps and for computing parts of the gradient. In the following, the rendering of the shape is displayed with *flat* shading on the facets.

Initialization The initial calibration $\mathbf{\Pi}^0$ is assumed to be given, either by a pre-calibrated multi-view setup, or after classical structure-from-motion. For the geometry estimation, we first apply the same algorithm described in this paper without the camera parameters updates. Then the *PBA* is performed by optimizing all parameters, $\mathbf{X}$, $\mathbf{\Pi}$ and $\mathbf{T}$.

Texture mapping and super-resolution In order to efficiently handle the texture, a texture atlas is generated. This allows easier access to neighboring texture values of a given point in order to easily compute gradients over the texture.

First, a labeling of each facet based on camera visibility information is computed, by finding the best frontal camera. In order to favor larger texture segments, a graph cut is performed on the mesh using alpha expansion. This gives a facet to camera mapping. The visible facets are projected on the corresponding images, and those coordinates are used for the texture mapping. This allows us to obtain a texture sampling coherent with the image resolution. Texture intensity values are computed by using Equation (13).

During the optimization, a coarse to fine strategy is also used in order to avoid local minima. This includes dealing with the resolution of the images, the resolution of the mesh and the resolution of the estimated texture. We make sure that the resolution of the texture is higher than the sampling over each triangle so that we have at least a few residual samples per triangle (the texture sampling is consistent

with the input image resolution by construction of the texture map). By choosing a higher texture resolution, we can improve the texture estimation. This actually corresponds in performing super-resolution with a simple bilinear convolution kernel which is naturally handled in our case. We could straight-forwardly extend this concept to arbitrary convolution kernels [9] without changing our approach.

Remeshing In order to avoid self-intersections during the mesh optimization, we use the topology-adaptive meshes described in [16]. Keeping track of the texture optimization while remeshing is rather difficult. To simplify the algorithm, the texture is reinitialized after remeshing using Equation (13). In practice we remesh every 20 iterations.

4.1. Pose and Shape Estimation

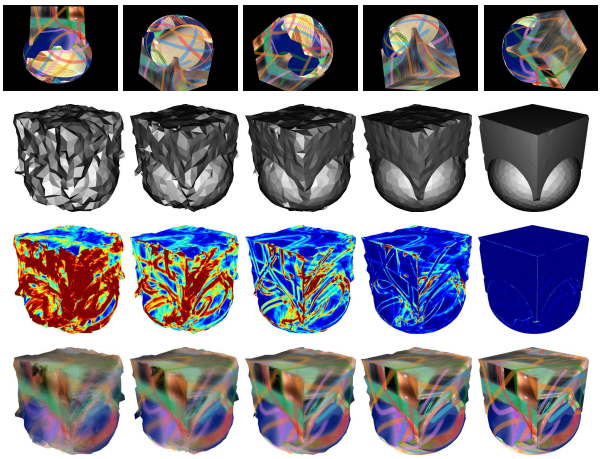

Figure 3. Evolution of the minimization at different iterations on a synthetic data (24 images of 640×480). From top to bottom: 5 of the input images; 3D triangular mesh $\mathbf{X}$; Photometric reprojection error; Textured mesh using the estimated texture $\mathbf{T}$.

We first evaluate our approach on synthetic data. It consists of 24 images of an imbricated cube and ball with Lambertian texture. We add Gaussian noise on the camera poses. We first run the baseline method without the camera updates (standard multi-view stereo) from the noisy data (starting from a simple sphere), and then we run the *PBA*. See Figure 3 for results. In this experiment, the algorithm converges in 300 iterations in about 30 minutes. Note that minimizing a discrete energy over a triangular mesh with a coherent gradient descent flow allows vertices to move in their correct location, and allows to preserve sharp edges which most previous methods are not able to achieve.

We evaluated on the classical Middlebury Temple and Dino datasets (See Figure 4). The accuracy is improved in both datasets, showing the advantage of jointly estimating the geometry and the calibration. Our approach is comparable to the state-of-the-art, and visually looks more appealing than some of the best methods as some details are nicely visible. For example holes in the columns are correctly reconstructed where many methods tend to oversmooth the surface. Similarly as described in [1, 5], we observe an improvement in both accuracy and completeness due to the camera refinement (See results Table 1). Fig 5 shows an additional results of a statue in the Rietberg Museum, Zurich, where we initialize the camera calibration with structure-from-motion [27, 26] and compute an initial mesh via [12]. Then we use the *PBA* approach described in this paper.

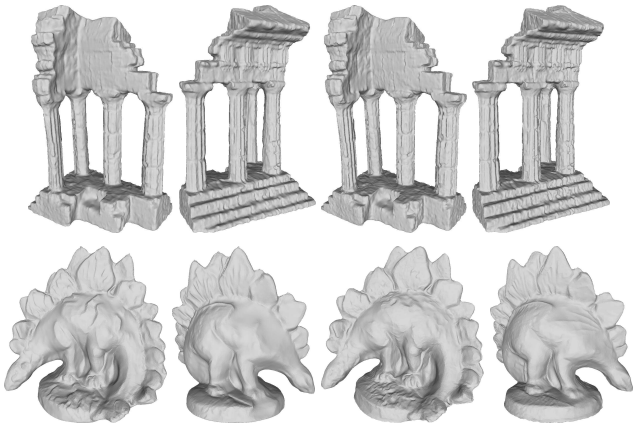

Figure 4. Results on the Middlebury stereo benchmark [18]. for DinoRing and TempleRing data (47 images) before (Left) and after (Right) the refinement with the proposed *PBA* method.

Figure 5. "Seated Bodhisattva" (50 images). Textured (mid.) and shaded (right) reconstructed surface with the proposed refinement.

	Temple Sparse Ring (16 images)			Temple Ring (47 images)			Dino Ring (48 images)		
	accu.(mm)	compl.(%)	photo.err.	accu.(mm)	compl.(%)	photo.err.	accu.(mm)	compl.(%)	photo.err.
Baseline	0.78	96.2	5.3238	0.59	99.0	5.8669	**0.51**	97.2	1.7154
Proposed PBA	**0.7**	**96.6**	**3.6024**	**0.51**	**99.1**	**3.7566**	**0.51**	**98.7**	**1.0863**

Table 1. Numerical evaluation of the proposed method for Middlebury Dino and Temple data sets [18] (Baseline: Multi-view stereo without calibration; Proposed: Same as before *with* calibration - Photometric Bundle Adjustment)). The table shows accuracy at 90% and completeness at 1.25mm, and *err*, the mean photometric reprojection error (in term of intensity values).

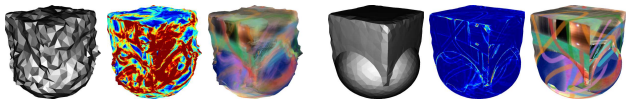

Figure 6. Results on *PBA* with both intrinsics and extrinsics refinement (estimated mesh, reprojection error and estimated texture, respectively). Left: Initial shape estimated without calibration refinement; Right: result of our *PBA* method.

4.2. Intrinsics Estimation

Similarly as before, we used the same simulated data and added a Gaussian noise to all camera parameters, this time including the intrinsics as well (focal length and principal point). Results are shown in Figure 6.

Figure 7 shows results on a publicly available dataset [11], were we tested our *PBA* using all parameters. While the intrinsic calibration does not change for most of the images, two views (namely #9 and #20) have a particularly wrong focal length. For the bird data, before the camera refinement with the baseline method (see Figure 8), the reprojection error is 3.2707 (average error per point on the surface in term of intensity values). After the photometric bundle adjustment, the reprojection error drops to 2.1359.

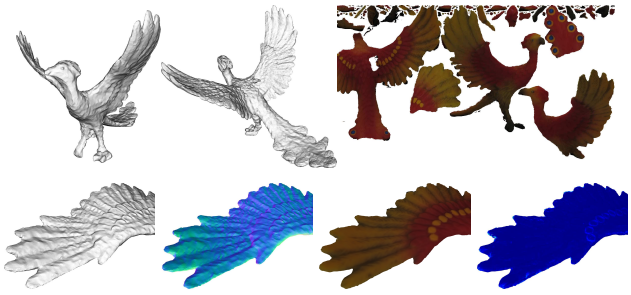

Figure 7. Bird shape (21 images) from [11]. Top: reconstructed surface and texture map. Bottom: details (3D geometry, color coded surface normals, textured mesh, reprojection error).

Perspectives

Even though the minimized error is rather simple (per pixel squared error), we are able to achieve high quality reconstructions comparable to previous techniques using more robust cost measures. While some parts of the surface contain flaws, we believe those problems come from the presence of local minima mostly due to non-Lambertian surfaces or matching ambiguities. A robust image similarity measure ([23]), or taking more parameters in the camera calibration (geometric distortions, radiometric models) would probably improve the reconstruction. However it is clear that the reprojection error is reduced showing significant improvements on the reconstructed surface and texture.

5. Conclusion

A dense image-based photometric bundle adjustment is presented, minimizing the reprojection error between a generated image and an observed image. The error is a simple image error motivated by a Bayesian vision of the multi-view reconstruction problem. It jointly refines the geometry (mesh) and calibration, leading to notable improvements both in the reconstructed geometry and the estimated texture on several datasets. The discrete gradient descent flow allows vertices to be moved at their correct location and to preserve surface edges (as in [2, 23]). This paper is a first and necessary step towards full dense multi-view bundle adjustment problems dealing with more complete generative models such as convolution or radiometry (reflectance, illumination), and can straight-forwardly be applied to any generative approaches dealing with multi-view reconstruction settings minimizing reprojection errors (i.e. multi-view range maps integration, multi-view photometric stereo).

Acknowledgements We gratefully acknowledge the support of the 4DVideo ERC Starting Grant Nr. 210806. We would like to thank [11, 18] for sharing data, and in particular D. Scharstein for the Middlebury evaluation [18].

References

[1] M. Aubry, K. Kolev, B. Goldluecke, and D. Cremers. Decoupling photometry and geometry in dense variational camera calibration. In *ICCV*, 2011. 2, 6

[2] A. Delaunoy and E. Prados. Gradient flows for optimizing triangular mesh-based surfaces: Applications to 3d reconstruction problems dealing with visibility. *IJCV*, 2011. 2, 3, 4, 5, 7, 8

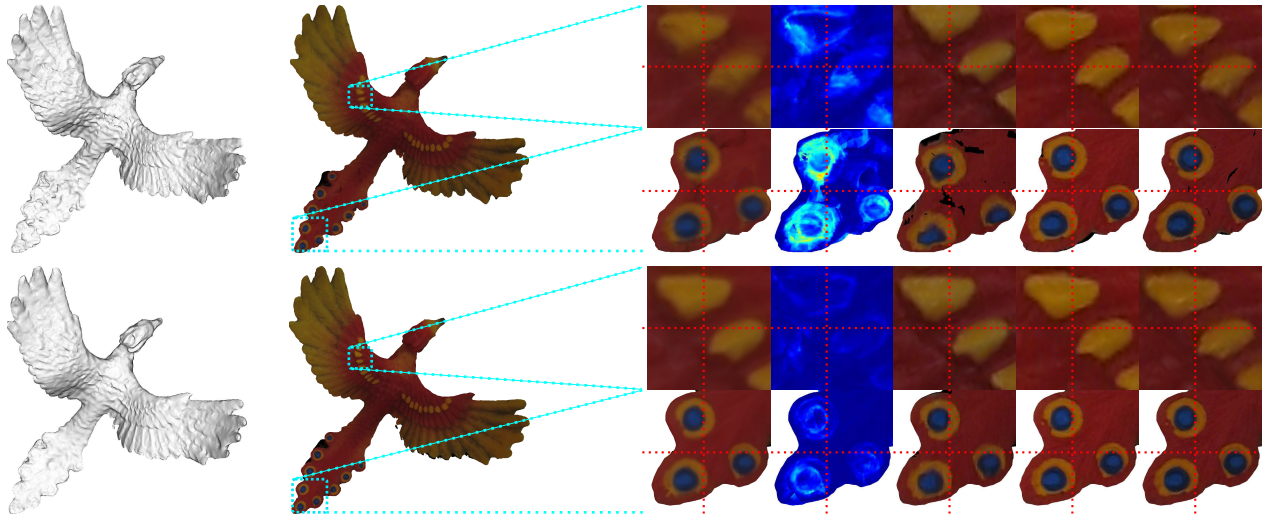

Figure 8. Comparison between [2] (Top row) and our approach (Bottom row) with camera refinement (both intrinsics and extrinsics) in the *PBA*. Close-up details (from left to right): Estimated texture; Photometric reprojection error (averaged over the mesh by summing over all images); Reprojection of image #0, #1 and #9 respectively.

[3] O. Faugeras and R. Keriven. Variational-principles, surface evolution, pdes, level set methods, and the stereo problem. *TIP*, 1998. 2

[4] A. Fitzgibbon. Simultaneous linear estimation of multiple view geometry and lens distortion. In *CVPR*, 2001. 1

[5] Y. Furukawa and J. Ponce. Accurate camera calibration from multi-view stereo and bundle adjustment. *IJCV*, 2009. 2, 6

[6] P. Gargallo. *Contributions to the Bayesian approach to Multi-view Stereo*. PhD thesis, 2008. 3

[7] P. Gargallo, E. Prados, and P. Sturm. Minimizing the reprojection error in surface reconstruction from images. In *ICCV*, 2007. 2, 3, 4, 5

[8] P. Georgel, S. Benhimane, and N. Navab. A unified approach combining photometric and geometric information for pose estimation. In *BMVC*, 2008. 2

[9] B. Goldluecke, M. Aubry, K. Kolev, and D. Cremers. A super-resolution framework for high-accuracy multiview reconstruction. *IJCV*, 2014. 2, 3, 4, 5, 6

[10] K. Kolev, M. Klodt, T. Brox, and D. Cremers. Continuous global optimization in multiview 3d reconstruction. *IJCV*, 2009. 2

[11] K. Kolev, T. Pock, and D. Cremers. Anisotropic minimal surfaces integrating photoconsistency and normal information for multiview stereo. In *ECCV*, 2010. 7

[12] P. Labatut, J.-P. Pons, and R. Keriven. Efficient multi-view reconstruction of large-scale scenes using interest points, delaunay triangulation and graph cuts. In *ICCV*, 2007. 6

[13] M. A. Lourakis and A. Argyros. SBA: A Software Package for Generic Sparse Bundle Adjustment. *ACM Trans. Math. Software*, 2009. 1

[14] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. Dtam: Dense tracking and mapping in real-time. In *ICCV*, 2011. 2

[15] M. Pollefeys, R. Koch, and L. Van Gool. Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters. *IJCV*, 1999. 1

[16] J. Pons and J. Boissonnat. Delaunay deformable models: Topology-adaptive meshes based on the restricted delaunay triangulation. In *CVPR*, 2007. 6

[17] J.-P. Pons, R. Keriven, and O. Faugeras. Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. *IJCV*, 2007. 2

[18] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *CVPR*, 2006. 2, 6, 7

[19] S. Soatto, A. J. Yezzi, and H. Jin. Tales of shape and radiance in multi-view stereo. In *ICCV*, 2003. 2, 3

[20] C. Strecha, W. V. Hansen, L. J. V. Gool, P. Fua, and U. Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *CVPR*, 2008. 2

[21] B. Triggs, P. Mclauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment – a modern synthesis. In *International Workshop on Vision Algorithms*, 2000. 1

[22] G. Unal, A. Yezzi, S. Soatto, and G. Slabaugh. A variational approach to problems in calibration of multiple cameras. *PAMI*, 2007. 2, 3, 4

[23] H.-H. Vu, P. Labatut, J.-P. Pons, and R. Keriven. High accuracy and visibility-consistent dense multiview stereo. *PAMI*, 2012. 2, 4, 5, 7

[24] A. Yezzi and S. Soatto. Stereoscopic segmentation. *IJCV*, 2003. 3

[25] S. Yezzi, A. J.and Soatto. Structure from motion for scenes without features. In *CVPR*, 2003. 2

[26] C. Zach and Others. V3D. `http://www.inf.ethz.ch/personal/chzach/opensource.html`. 6

[27] C. Zach and M. Pollefeys. Practical methods for convex multi-view reconstruction. In *ECCV*. 2010. 6