

# Large Scale Photometric Bundle Adjustment

Oliver J. Woodford  
o.j.woodford.98@cantab.net  
Edward Rosten

Snap, Inc., Santa Monica.  
Snap Group Ltd., London.

---

## Abstract

Direct methods have shown promise on visual odometry and SLAM, leading to greater accuracy and robustness over feature-based methods. However, offline 3-d reconstruction from internet images has not yet benefited from a joint, photometric optimization over dense geometry and camera parameters. Issues such as the lack of brightness constancy, and the sheer volume of data, make this a more challenging task. This work presents a framework for jointly optimizing millions of scene points and hundreds of camera poses and intrinsics, using a photometric cost that is invariant to local lighting changes. The improvement in metric reconstruction accuracy that it confers over feature-based bundle adjustment is demonstrated on the large-scale Tanks & Temples benchmark. We further demonstrate qualitative reconstruction improvements on an internet photo collection, with challenging diversity in lighting and camera intrinsics.

## 1 Introduction

The joint estimation of camera parameters and scene structure from a set of images is a fundamental Computer Vision problem, with applications from online camera pose estimation for augmented reality, to large scale reconstruction of objects, buildings and cities for mapping, game asset generation and historical archiving. The former, visual odometry task has recently been shown [8] to significantly improve in accuracy when using a photometric error, rather than the geometric error of more traditional, feature-based methods. There are good theoretical reasons for this: these “direct” methods optimize in the domain of pixel errors, the true source of measurement noise. In addition, the approach requires localizability in only 1-d, along epipolar lines, rather than 2-d for feature-based methods. This enables the use of intensity *edges*, in addition to corners, allowing for a denser reconstruction, and thus more constraints on camera parameters also.

Despite these advantages, the latter task of large scale reconstruction, in particular from sets of internet images, consisting of a large number of photos, each with their own camera intrinsics and lighting conditions, has not yet benefited from a joint, photometric treatment. The de-facto standard approach to this task is to compute camera parameters and sparse geometry using a feature-based structure from motion (SfM) method [2], followed by dense geometry reconstruction using a multi-view stereo (MVS) method. The goal of this work is to bring the benefits of a photometric error to the first stage, joint camera and structure estimation, improving the accuracy of inputs to an MVS second stage. Specifically, we

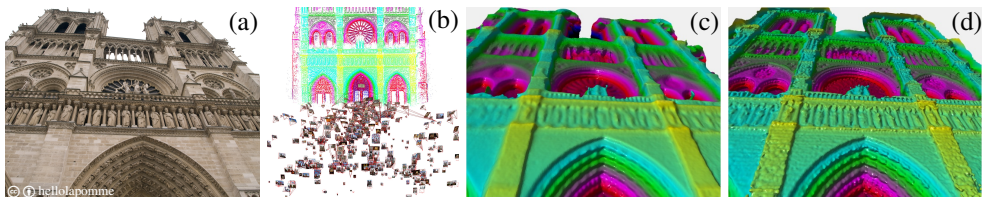


Figure 1: Given 700+ photos of Notre Dame (e.g. a), captured with different cameras and lighting conditions, our method refines the camera poses (b, red), intrinsics, and dense geometry (c) produced by a standard SfM+MVS framework [25, 26], using a joint, photometric optimization. Both the new poses (b, black) and 3-d landmarks (b) can be used to generate higher fidelity dense reconstructions of the scene, e.g. via Poisson meshing [18] (d).

tackle the problem of large scale, photometric bundle adjustment, *i.e.* the joint refinement of camera and structure parameters under a photometric error, addressing two key challenges:

1. Handling the variety of both lighting conditions and intrinsic parameters present in a large and diverse set of source images, such as those downloaded from the internet.
2. Solving an optimization problem involving thousands of camera variables and millions of geometry variables in an efficient and effective manner.

We do not tackle the initialization problem, required for a fully photometric SfM pipeline, instead using off-the-shelf software to generate initial parameters. However, we demonstrate that even a photometric refinement of feature-based estimates yields a significant improvement in reconstruction accuracy, as demonstrated quantitatively using the Tanks and Temples (TT) benchmark [19], and qualitatively on an internet photo collection.

## 2 Related work

The joint optimization of structure and camera parameters is common within feature-based systems [22], which minimize a geometric error. Often the feature locations themselves are a result of a photometric optimization (e.g. KLT tracking [21]). Alternating optimizations of such geometric and photometric errors yields improvements [9]. However, few methods exist which minimize a photometric error directly over structure and motion. These fall into two main categories: offline reconstruction [6, 12], and visual odometry (VO) [2, 3, 8, 17].

The offline methods [6, 12] model scene structure densely with triangulated meshes, regularized for smoothness. A texture map is inferred, using a texture-to-image error, allowing appearance to be super-resolved [12]. This significantly increases both the number of variables (due to the texture) and dependence between them (due to the mesh and smoothness regularization). As a result, optimization is either alternated over different sets of variables (texture, structure, cameras) [12], or a simple, first order, gradient descent solver [6].

The VO methods minimize an image-to-image error [2, 3, 8, 17, 23], using the structure to compute correspondences between images, thus avoiding the need to infer texture. Image-to-image errors require handling both lens distortion and inverse, or un-, distortion. Camera intrinsics are assumed known and fixed, thus avoiding optimizing lens parameters through the undistortion process. Most methods [2, 3, 8, 23] model structure with sparse, ray-based landmarks: fronto-parallel patches anchored to a pixel location in a source frame, with variable depth. Some MVS methods [11, 12] optimize both the depth and normal of

landmarks, though not jointly with camera parameters. Similarly, earlier photometric VO work [14] tracks a few planes of broad extent, optimizing both plane parameters and camera extrinsics. Without smoothness regularization, the landmarks or planes of these VO methods [2, 3, 8, 14] are independent of each other. The methods do joint optimization using second order solvers, improving the speed of convergence, but on relatively small problems.

Most of these methods assume constant brightness of a scene point in all images [2, 3, 8, 14, 15]. Non-Lambertian surfaces or lighting changes due to time of day or year, or a shifting light source, or by images taken with different cameras, invalidate this assumption. Alismail *et al.* [9] transform images into an 8-channel, lighting invariant, binary feature space prior to minimizing the photometric error; this makes the method invariant to local lighting changes, at a cost to computation time and convergence basin size [30]. Park *et al.* [23] evaluated this and other approaches to illumination robustness in the context of direct SLAM. MVS frameworks often use the Normalized Cross Correlation (NCC) photometric score [10, 26], which is invariant to affine intensity variations, over local patches. Recent work on image alignment [30] has incorporated this measure into a standard, least squares optimization framework, employed here.

## 2.1 Our contributions

Despite computing dense geometry, our approach has more in common with the VO approaches mentioned above, using an independent, ray-based, planar landmark representation for structure, and a joint, second order solver for optimization. We contribute the following:

1. Applying an NCC-based photometric framework [30] to bundle adjustment. While this measure has been applied to both tracking and MVS, it has not been optimized jointly over both structure and camera parameters.
2. Optimizing lens distortion parameters with image-to-image errors, requiring differentiation through the lens undistortion process.
3. A memory efficient implementation of the Variable Projection optimizer [13, 16], enabling the joint optimization of thousands of camera parameters and millions of structure parameters on a desktop PC.

## 3 Method

We now describe the key components of our framework: parameterization of camera and structure variables, the photometric cost function, and the optimization framework, plus additional implementation details.

### 3.1 Problem parameterization

Camera parameters define the projection of a 3-d point,  $\mathbf{X} \in \mathbb{R}^3$ , in world coordinates, onto the image plane, in pixel coordinates. Camera extrinsics consist of  $P$  world to image rotations and translations,  $\{\mathbf{R}_i, \mathbf{t}_i\}_{i=1}^P$ ,  $\mathbf{R}_i \in \mathbf{SO}(3)$ ,  $\mathbf{t}_i \in \mathbb{R}^3$ , one pair per *image*. Intrinsic consist of  $C$  linear and lens calibration parameters,  $\{\mathbf{s}_j, \mathbf{I}_j\}_{j=1}^C$ ,  $\mathbf{s}_j \in \mathbb{R}^4$ ,  $\mathbf{I}_j \in \mathbb{R}^2$ , one pair per *camera*, where  $C \leq P$ . When  $C < P$ , some camera intrinsics are shared across input images; in this case an index mapping from image  $i$  to camera  $j$  is required as input. To simplify notation,

we hide this mapping where necessary, and refer to both extrinsics and intrinsics of a given image using the same index. The world to pixel coordinate ( $\mathbf{x}'$ ) transform is then given by

$$\mathbf{x}' = \kappa_{s_j}(\varphi_{l_j}(\pi(\mathbf{R}_j \mathbf{X} + \mathbf{t}_j))), \quad (1)$$

where  $\pi(\cdot) : \mathbb{R}^3 \rightarrow \mathbb{R}^2$  is the projection function  $\pi([x, y, z]^T) = [x/z, y/z]^T$ ,  $\varphi_l(\cdot) : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  is a lens distortion function, and  $\kappa_s(\cdot) : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  is the linear calibration function

$$\kappa_s(\mathbf{x}) = \begin{bmatrix} s_1 & 0 \\ 0 & s_2 \end{bmatrix} \mathbf{x} + \begin{bmatrix} s_3 \\ s_4 \end{bmatrix}, \quad \text{s.t.} \quad \kappa_s^{-1}(\mathbf{x}) = \begin{bmatrix} 1/s_1 & 0 \\ 0 & 1/s_2 \end{bmatrix} \left( \mathbf{x} - \begin{bmatrix} s_3 \\ s_4 \end{bmatrix} \right). \quad (2)$$

For lens distortion, we use a standard polynomial radial distortion model:

$$\varphi_l(\mathbf{x}) = \mathbf{x}(1 + l_1 r + l_2 r^2 + \dots + l_n r^n), \quad \text{where} \quad r = \|\mathbf{x}\|^2. \quad (3)$$

For world to camera distortion,  $n = 2$  and  $[l_1, l_2] = \mathbf{l}_j$ . For camera to world undistortion (required for our ray-based formulation, described below), the same model can be used with a different set of polynomial coefficients representing the inverse transformation, s.t.  $\varphi_l^{-1}(\mathbf{x}) = \varphi_{\phi(\mathbf{l})}(\mathbf{x})$ . We compute the undistortion coefficients,  $\phi(\mathbf{l})$ , in closed form using the first six coefficient formulae of Drap & Lefèvre [DL, Appendix C].

We use a ray-based parameterization of structure, whereby each landmark is anchored to a pixel in an input image. Since we are comparing image texture around such points, we avoid making assumptions about the normal of the surface, and instead model it explicitly. Each landmark consists of a given (fixed) pixel location  $\mathbf{x}$ , source frame index  $i$ , and the variable surface plane parameterization  $\mathbf{n} \in \mathbb{R}^3$  of Habbecke & Kobbelt [HK], in the source image coordinate frame. Its world coordinates are then computed as follows:

$$\mathbf{X} = \mathbf{R}_i^T \left( \frac{\bar{\mathbf{x}}}{\mathbf{n}^T \bar{\mathbf{x}}} - \mathbf{t}_i \right), \quad \text{where} \quad \bar{\mathbf{x}} = \begin{bmatrix} \varphi_{l_i}^{-1}(\kappa_{s_i}^{-1}(\mathbf{x})) \\ 1 \end{bmatrix}. \quad (4)$$

A pixel to pixel correspondence  $\mathbf{x} \rightarrow \mathbf{x}'$  from source frame  $i$  to target frame  $j$ , for landmark  $k$ , can thus be achieved through the substitution of equation (4) into equation (1), which we represent, for  $N$  image coordinates, constituting a patch around the landmark, with the function  $\Pi_{ijk}^\Theta(\cdot) : \mathbb{R}^{2 \times N} \rightarrow \mathbb{R}^{2 \times N}$ .  $\Theta = \{\{\mathbf{R}_i, \mathbf{t}_i\}_{i=1}^P, \{\mathbf{s}_j, \mathbf{l}_j\}_{j=1}^C, \{\mathbf{n}_k\}_{k=1}^L\}$  denotes the set of all problem parameters, the variables to be optimized.  $L$  is the number of landmarks.

### 3.1.1 Update parameterization

Each iteration of optimization computes a parameter update,  $\delta\Theta$ . Most parameters, with the exception of rotations, minimally parameterize a Euclidean space, therefore are updated additively, e.g.  $\mathbf{n}_k \leftarrow \mathbf{n}_k + \delta\mathbf{n}_k$ . For rotations, the update is parameterized (minimally) as  $\mathbf{R}_i \leftarrow \mathbf{R}_i \Omega(\delta\mathbf{r}_i)$ , where  $\Omega(\cdot)$  is Rodrigues' formula [R] for converting a 3-vector into a rotation matrix. In an abuse of notation, by a derivative of rotation,  $\frac{\partial}{\partial \mathbf{R}}$ , we mean the derivative of the update,  $\frac{\partial}{\partial \delta \mathbf{r}} \Big|_{\delta \mathbf{r}=0}$ . The update of parameters in general is denoted  $\Theta \leftarrow \Theta \oplus \delta\Theta$ .

## 3.2 Cost formulation

Our parameterization gives us a mapping from pixels in one image to pixels in another, via the scene geometry and camera positions and intrinsics; our cost should measure the

difference between those two sets of pixels. To ensure that our cost is invariant to local lighting changes as well as unexpected occlusions, we use a robust, locally normalized, least squares NCC cost [60]. Specifically, for each landmark (indexed by  $k$ ), anchored in image  $\mathbf{I}_i \in \mathbb{R}^{H \times W}$  (we use grayscale images), where  $i = I_k$  is the source image index of the  $k^{\text{th}}$  landmark, we define a  $4 \times 4$  patch of pixels centered on it, with the set of image coordinates  $\mathbf{P}_k \in \mathbb{R}^{2 \times N}$  ( $N = 16$ ). Each landmark is visible in a subset of input frames, the (given) set of indices of which is denoted  $\mathcal{V}_k$ . The cost over all landmarks and images is thus given by

$$\text{Total cost:} \quad E(\Theta) = \sum_k \sum_{j \in \mathcal{V}_k} \rho(\|\mathcal{E}_{jk}\|^2), \quad \rho(s) = \frac{s}{s + \tau^2}, \quad (5)$$

$$\text{Patch residual:} \quad \mathcal{E}_{jk} = \Psi\left(\mathbf{I}_j \left(\Pi_{ijk}^\Theta(\mathbf{P}_k)\right)\right) - \Psi(\mathbf{I}_i(\mathbf{P}_k)), \quad i = I_k, \quad (6)$$

$$\text{NCC normalization:} \quad \Psi(\bar{\mathbf{I}}) = \frac{\bar{\mathbf{I}} - \mu_{\bar{\mathbf{I}}}}{\sigma_{\bar{\mathbf{I}}}}, \quad \mu_{\bar{\mathbf{I}}} = \frac{\mathbf{1}^T \bar{\mathbf{I}}}{N}, \quad \sigma_{\bar{\mathbf{I}}} = \|\bar{\mathbf{I}} - \mu_{\bar{\mathbf{I}}}\|, \quad (7)$$

with  $\mathbf{I}(\mathbf{P}) = \bar{\mathbf{I}}$  representing sampling, and  $\mathbf{1}$  denoting a vector of ones. The Geman-McClure kernel [5, 60],  $\rho$ , robustifies costs with  $\tau = 0.5$ . The source frame,  $I_k$ , can be ignored in  $\mathcal{V}_k$ , since it contributes no error, by construction.

### 3.3 Cost optimization

Equation (5) defines a robustified non-linear least squares cost, for which many solvers exist [24]. These generally involve computing the partial derivatives of residual errors,  $\mathcal{E}$ , w.r.t. to the optimization variables, known as the Jacobian,<sup>1</sup>  $\mathbf{J} = \frac{\partial \mathcal{E}}{\partial \Theta}$ . Standard implementations of such solvers, *e.g.* Ceres Solver [4], cache the whole Jacobian, which would be close to 3TB for one dataset used here. It is not surprising that some approaches resort to alternative strategies to optimize this problem [6, 12].

However, our problem has a special structure, common to BA: without surface regularization, the landmarks are independent of each other. Enter the Variable Projection (VarPro) method [13, 16], that, using the Schur complement, allows us to construct and solve a small Reduced Camera System (RCS) problem, then solve for the structure using Embedded Point Iterations (EPIs). The RCS involves the set of all problem variables excluding structure variables, which we denote  $\bar{\Theta}$ . The RCS is constructed and solved, using Levenberg-style damping [20], as follows [16]:

$$\delta \bar{\Theta} = -(\mathbf{H}_{\text{rCS}} + \mathbf{J}_{\text{reg}}^T \mathbf{J}_{\text{reg}} + \lambda \mathbf{I})^{-1} (\mathbf{g}_{\text{rCS}} + \mathbf{J}_{\text{reg}}^T \mathcal{E}_{\text{reg}}), \quad (8)$$

$$\mathbf{H}_{\text{rCS}} = \sum_{k=1}^L \bar{\mathbf{J}}_k^T (\mathbf{I} - \hat{\mathbf{J}}_k \hat{\mathbf{J}}_k^+) \bar{\mathbf{J}}_k, \quad \mathbf{g}_{\text{rCS}} = \sum_{k=1}^L \bar{\mathbf{J}}_k^T (\mathbf{I} - \hat{\mathbf{J}}_k \hat{\mathbf{J}}_k^+) \mathcal{E}_k, \quad (9)$$

$$\mathcal{E}_k = [\rho'(\mathcal{E}_{jk}) \mathcal{E}_{jk}]_{\forall j \in \mathcal{V}_k}, \quad \rho'(s) = \frac{\partial}{\partial s} \rho(s) = \frac{\tau^2}{(s + \tau^2)^2} \quad (10)$$

$$\bar{\mathbf{J}}_k = \left[ \rho'(\mathcal{E}_{jk}) \frac{\partial \mathcal{E}_{jk}}{\partial \bar{\Theta}} \right]_{\forall j \in \mathcal{V}_k}, \quad \hat{\mathbf{J}}_k = \left[ \rho'(\mathcal{E}_{jk}) \frac{\partial \mathcal{E}_{jk}}{\partial \mathbf{n}_k} \right]_{\forall j \in \mathcal{V}_k}, \quad (11)$$

$$\mathcal{E}_{\text{reg}} = 10^5 \cdot \left[ \begin{array}{ccc} s_{1i-s_{2i}} & & \\ s_{1i+s_{2i}} & s_{3i} & s_{4i} \end{array} \right]_{\forall i \in \{1, \dots, C\}}^T, \quad \mathbf{J}_{\text{reg}} = \frac{\partial \mathcal{E}_{\text{reg}}}{\partial \bar{\Theta}}, \quad (12)$$

<sup>1</sup>Formulae for specific Jacobians of our cost function are not presented. They can be derived straightforwardly, but modern auto-differentiation tools, such as the C++ Jet type [4] (employed here), make implementing these formulae unnecessary.

where  $\mathbf{J}^+ = (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T$ , denoting the matrix pseudo-inverse,  $\mathbf{I}$  is the identity matrix, and  $\lambda$  is the damping parameter.  $\mathcal{E}_{\text{reg}}$  is a regularization term that ensures linear camera intrinsics are well constrained. Following the camera parameter update, EPIs are run using a Gauss-Newton update:

$$\delta \mathbf{n}_k = -\hat{\mathbf{J}}_k^+ \mathcal{E}_k, \quad (13)$$

until convergence. Our implementation groups Jacobians per landmark, and sums the reduced system over landmarks (eq. (9)). While mathematically equivalent to standard VarPro [16, 23], this explicitly orders the Jacobian computations. Both the EPIs and construction of the RCS can thus be run over each landmark independently, in parallel. Jacobians for each landmark are not referenced outside these computations, therefore we do not store Jacobians beyond each iteration of the loops over landmarks, slashing the memory requirements of this method.<sup>2</sup> To limit computation time, VarPro is stopped after just ten iterations. The full optimization is described in Algorithm 1.

### 3.4 Initialization, and other implementation details

The framework presented here jointly refines structure and camera parameters of an existing reconstruction, to improve accuracy. Off-the-shelf SfM + MVS systems can provide an initial  $\Theta$ . In particular, we use COLMAP [24] with out-of-the-box parameters<sup>3</sup> to produce initial camera parameters (via SfM [25]) and landmark parameters (via MVS [26]).<sup>4</sup> Refining *dense* structure jointly with camera parameters in this way might not be needed in some applications, but serves to demonstrate what is feasible with our low memory formulation. Unnecessary background points slow computation, so we manually select landmarks roughly on the object of interest.

In addition to the camera and landmark parameters, our framework needs a source frame index,  $I_k$ , and visibilities,  $\mathcal{V}_k$ , per landmark; these remain fixed throughout the optimization. We perform a Poisson surface reconstruction [18, 24] on the selected landmarks, and use the resulting mesh to compute visibilities: the mesh is rendered into each view as a depth map, landmarks are projected into the view, and their depths compared to the depth map; those that differ by  $< 1\%$  are deemed visible. In order to avoid selecting a source frame that is a photometric outlier (*e.g.* due to a specularity),  $I_k$  is chosen as the frame whose patch is closest to a robust mean of the visible, normalized patches:

$$I_k = \underset{j \in \mathcal{V}_k}{\operatorname{argmin}} \|\bar{\mathbf{I}}_j - \boldsymbol{\mu}\|^2, \quad \boldsymbol{\mu} = \underset{\hat{\boldsymbol{\mu}} \in \mathbb{R}^N}{\operatorname{argmin}} \sum_{j \in \mathcal{V}_k} \rho(\|\bar{\mathbf{I}}_j - \hat{\boldsymbol{\mu}}\|^2), \quad (14)$$

$$\bar{\mathbf{I}}_j = \Psi(\mathbf{I}_j(\kappa_{s_j}(\varphi_j(\pi(\mathbf{R}_j \mathbf{X}_k + \mathbf{t}_j))))), \quad (15)$$

where  $\mathbf{X}_k$  is a  $3 \times N$  matrix of world coordinates, a  $4 \times 4$  grid of points, spaced such that the mean spacing in visible views is 1 pixel, on the plane around the  $k^{\text{th}}$  landmark.  $\boldsymbol{\mu}$  is computed using iteratively reweighted least squares [15], starting from the unrobustified mean. To ensure landmarks are only initialized in textured image regions, we remove ones for which  $\|\bar{\mathbf{I}}_{I_k}\| < 0.5N$  (assuming 256 gray levels).

<sup>2</sup>Previous VarPro bundle adjustment methods [16, 23] do not provide an explicit Jacobian ordering, therefore cannot exploit this memory reduction. Note that this low memory VarPro can be applied to all bundle adjustments, not just our photometric one.

<sup>3</sup>colmap automatic\_reconstructor, with TT datasets using `-single_camera`.

<sup>4</sup>COLMAP outputs the position and normal direction for each landmark, from which our landmark parameterization,  $\mathbf{n}_k$ , can be initialized.

Image pyramids are used to improve convergence. We run the optimization on half size source frames first, followed by full size. Furthermore, to reduce aliasing, target frames are sampled, using bilinear interpolation, at the image pyramid level which produces image samples that are closest to one pixel apart, for each residual  $\mathcal{E}_{jk}$ . Finally, we optimize structure alone prior to commencing joint optimization at the first resolution.

## 4 Evaluation

While we compute both scene geometry and camera poses, our metric of choice is reconstruction accuracy, rather than the camera position accuracy used by VO methods, since reconstruction is more often the end goal of batch methods. Furthermore, ground truth geometry is more readily available than camera poses on large scale datasets, such as Temples and Tanks (TT) [19].

We perform a quantitative evaluation of metric reconstruction accuracy (up to scale) using the TT benchmark [19], whose ground truth geometry was captured by LIDAR. We additionally use their training datasets to run an ablation study highlighting the impact of several elements of our framework. The TT sequences, captured as video from a single camera<sup>5</sup>, do not have the variety of lighting conditions and camera intrinsics of an internet-sourced dataset, therefore we also provide qualitative results on an internet photo collection.

Ours is the first *photometric* bundle adjustment method suitable for large, diverse image sets with unknown camera poses and intrinsics; previous approaches have all been feature-based. We therefore pick a baseline from that category: COLMAP (SfM [25] + MVS [26]). This method leads publicly available, complete SfM + MVS pipelines on TT in terms of precision (our metric of interest), and it is the initializer for our method, so any difference in performance can be entirely attributed to our framework. Photometric bundle adjustment methods exist for more controlled scenarios [2, 3, 6, 8, 12, 17], but the VO methods [2, 3, 8, 17] cannot be applied to batches of images, while code is not available for existing batch methods [6, 12]. Nevertheless, our ablation study contrasts features of our framework with those of other photometric methods, so that our contributions can be fairly evaluated against those. In addition, direct comparisons can be done via the TT online leaderboard. We do not compare to state-of-the-art MVS methods, since they don't optimize camera parameters and also incorporate surface regularization and other prior knowledge.

### 4.1 Quantitative precision scores on TT

We ran our algorithm (LSPBA) on the TT intermediate image sets, and also ran COLMAP-MVS [26] using the camera parameters produced by our method (LSPBA + COLMAP-MVS), and submitted both sets of results to the online leaderboard [19]. The resulting scores<sup>6</sup> are presented in Table 1, along with those published for COLMAP<sup>7</sup>. Figure 3 visualizes the reconstructions, with colour encoding the distance from ground truth (lighter is closer).

The LSPBA method significantly improves the metric accuracy of reconstruction over COLMAP, improving the mean precision score by 21.8%. The recall score is 7% lower,

<sup>5</sup>For TT sequences we optimize a single, global set of camera intrinsics; for internet photo collections we optimize separate intrinsics for each image.

<sup>6</sup>Please refer to the TT paper [19] for details on how the scores are computed.

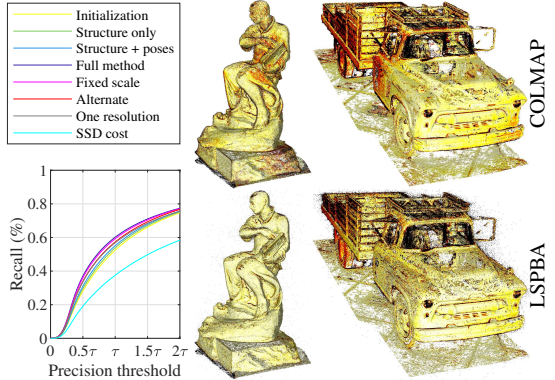
<sup>7</sup>TT COLMAP results may differ from the initial solutions used here, due to different settings, software versions, stochastic effects, and our culling of landmarks. COLMAP results given in §4.2 are our initialization (*i.e.* after landmark culling).

**Algorithm 1:** Low memory VarPro optimization

```

 $\lambda \leftarrow L$ ;  $\omega \leftarrow 10$ ; # Set damping parameters
Compute initial cost,  $S_0 \leftarrow E(\Theta_0)$ , (eq. (5));
for  $t = 1:10$  do
   $\Theta_t \leftarrow \Theta_{t-1}$ ;
  for  $k = 1:L$  do
    | Add landmark  $k$  to RCS (eq. (9))
  Compute cameras update (eq. (8));
   $\tilde{\Theta}_t \leftarrow \Theta_t \oplus \delta\Theta$ ;
  for  $k = 1:L$  do
    | while cost decreases do
      | Compute landmark  $k$  update (eq. (13));
      |  $\mathbf{n}_{kt} \leftarrow \mathbf{n}_{kt} + \delta\mathbf{n}_k$ ;
  Compute new cost,  $S_t \leftarrow E(\Theta_t)$ , (eq. (5));
  if  $S_t < S_{t-1}$  then
    |  $\lambda \leftarrow \lambda/10$ ;  $\omega \leftarrow 10$ ; # Reduce damping
  else
    |  $\Theta_t \leftarrow \Theta_{t-1}$ ;
    |  $\lambda \leftarrow \max(\lambda\omega, 10^{-6})$ ; # Increase damping
    |  $\omega \leftarrow 2\omega$ ;
  go to retry;

```



(a) Mean precision (b) Ignatius (c) Truck

Figure 2: Results on the TT training sets.

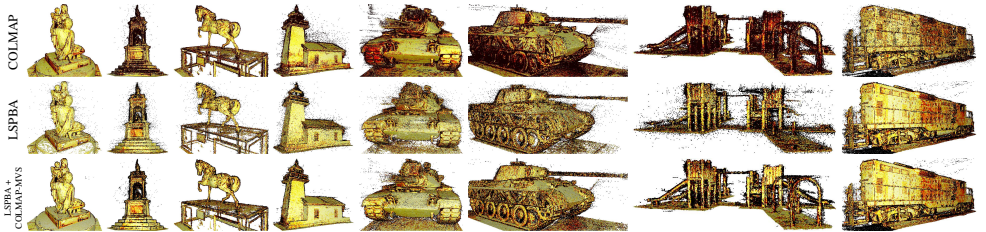


Figure 3: Precision error visualization for methods on the TT intermediate sets.

	Family	Francis	Horse	Light-house	M60	Panther	Play-ground	Train	Mean
COLMAP [25, 26]	56.02	34.35	40.34	53.51	41.07	39.94	38.17	41.93	43.16
(Baseline)	45.82	16.46	18.79	59.69	49.34	<b>57.01</b>	<b>66.61</b>	<b>42.15</b>	44.48
LSPBA	<b>68.76</b>	<b>55.79</b>	44.95	<b>61.91</b>	<b>50.48</b>	51.02	45.75	41.99	<b>52.58</b>
(Our method)	41.48	<b>26.82</b>	15.70	<b>64.43</b>	45.30	46.45	53.33	34.63	41.48
LSPBA +	66.15	44.60	<b>45.28</b>	57.16	50.36	<b>51.43</b>	<b>48.32</b>	<b>43.38</b>	50.84
COLMAP-MVS	<b>55.86</b>	23.48	<b>19.08</b>	57.71	<b>52.50</b>	56.15	64.44	30.35	<b>44.95</b>

Table 1: Published precision & recall scores for methods on the TT intermediate sets.

	Barn	Caterpillar	Church	Ignatius	Meeting-room	Truck	Mean	Change (%)
COLMAP output	38.00	34.79	50.04	60.39	39.50	50.96	45.61	0
Structure only	38.13	35.66	48.78	66.16	40.29	50.48	46.58	17.21
Structure + poses	41.18	35.47	47.41	69.90	40.41	50.71	47.51	33.79
Full method	<b>48.00</b>	<b>39.39</b>	49.42	72.61	<b>43.00</b>	<b>55.00</b>	<b>51.24</b>	100
Fixed scale	45.87	39.32	48.30	<b>72.63</b>	42.54	54.88	50.59	88.47
Alternate	40.50	38.41	<b>50.35</b>	72.54	41.61	52.15	49.26	64.84
One resolution	40.50	37.76	47.51	71.98	41.08	53.66	48.75	55.74
SSD cost	35.77	22.28	30.43	43.86	27.42	41.75	33.58	-214.0
Low qual. initial	38.35	29.76	51.91	50.63	37.26	47.55	42.58	-
Low qual. refined	52.48	41.80	60.42	71.88	44.16	58.18	54.82	-

Table 2: Precision AUC scores for various optimizations (discussed in §4.2) on the TT training sets. The final column shows the percentage increase in score of each method, relative to the increase of LSPBA (full method) over the baseline.



which is unsurprising given its lack of surface smoothness regularization; this allows some poorly constrained landmarks to leave the surface, particularly visible on the playground sequence. Nevertheless, recall is improved on two sequences. Running an MVS method using the refined camera parameters might be expected to give a similar improvement in accuracy, while maintaining the previous level of recall. This is exactly what LSPBA + COLMAP-MVS achieves; it improves accuracy over COLMAP on every sequence, by 17.8% on average, whilst slightly improving the average recall also.

## 4.2 Quantitative ablation study

In order to understand which elements of this framework provide benefit, we ran an ablation study on the TT training image sets, resulting in an error-recall curve for precision per sequence, the mean of which is shown in Figure 2(a), where  $\tau$  is the sequence dependent error threshold used in the TT benchmark. Also shown in Figure 2(b,c) are error visualizations for the COLMAP (top) and LSPBA (bottom) methods on two sets. The results are summarized in Table 2, by computing the area under each curve (per sequence), as a percentage of the total plot area. This AUC score captures more information than reporting recall at  $\tau$ , the value used in the TT benchmark. We describe and discuss each of the results below.

**Initialization** is the result of COLMAP, with textureless landmarks culled. It is the baseline, and starting point for all the other optimizations. Marginally better than LSPBA (full method) on the Church sequence, it is otherwise significantly worse.

**Structure only** is a two pyramid level optimization of structure parameters only, keeping camera poses and intrinsics fixed at their initial values. It delivers 17% of the improvement of the full method, on average, validating the need for a joint optimization.

**Structure + poses** is a two pyramid level, joint optimization of structure and pose parameters, keeping camera intrinsics fixed at their initial values. It provides 34% of the total improvement, validating the need to optimize camera intrinsics as well as poses.

**Full method** is the complete LSPBA method proposed here; a two pyramid level, joint optimization of structure and camera parameters. It achieves the best score on four of the six sequences, with a significant 12.3% improvement in AUC over COLMAP.

**Fixed scale** samples the target image pyramid at the same level as the source image pyramid, rather than using dynamic level selection. Very marginally best on Ignatius, it achieves 89% of the full method’s improvement, demonstrating the modest gains delivered by dynamic level selection.

**Alternate** replaces the RCS of VarPro with a camera system computed assuming structure is fixed. This then alternates camera and structure updates (10 times), similar to previous work [1, 2]. Marginally best on Church, this approach delivers 65% of the improvement of the full method overall, validating the benefit of VarPro over alternation.

**One resolution** applies LSPBA at only the largest image pyramid level, reducing the improvement to 56% of that using two pyramid levels, demonstrating the benefit of a coarse to fine approach.

**SSD cost** exchanges the NCC cost of the full method with the sum of squared differences (SSD) cost, which enforces the common constant brightness assumption [1, 3, 4, 5]. We used the Huber kernel as robustifier, with a transition threshold of  $40^2$ . This method significantly reduces the precision of the initial solution on all but one sequence, validating the need for a lighting invariant photometric cost in practical applications.

**Low quality** initial and refined rows refer to using COLMAP on the lowest quality set-

ting<sup>8</sup> for initialization, and refining this with LSPBA, respectively. Our method improves the accuracy of all sequences, with an average gain in AUC of 29% (much larger than for the standard initialization), suggesting that it extends well to other initializations.

### 4.3 Qualitative results on internet photo collections

Internet photo collections have a more diverse set of cameras and lighting conditions than the TT datasets, but lack ground truth data. We therefore present only qualitative results, on a publicly available dataset, “Notre Dame” [27], in Figure 1. Panel (b) shows the landmarks, coloured by relief depth, and camera positions before (red) and after (black) refinement. The lowest 10% of landmarks, ranked by mean photometric cost, are removed to filter out outliers. Comparing the filtered landmarks meshed using Poisson meshing [18] (d) with the COLMAP landmarks meshed similarly (c), our reconstruction captures significantly finer details, *e.g.* of arches on the towers. It does fail to fix existing, larger scale errors, such as missing balustrade, and introduces more noise on flat regions of the building, due to a lack of texture and smoothness regularization.

To give an idea of the computational resources required for our method, this photo collection, with 701 images and 755k landmarks, took about a week to optimize (not including COLMAP running time), using parallelized<sup>9</sup> C++ code on an 8 core 3.7GHz Xeon desktop PC, using 44GB of memory at peak; the full Jacobian for this problem would be 900GB. To accelerate experiments we used a 96 core 3GHz Xeon server; optimization of this dataset took under 5 hours on this machine. This would further improve with GPU acceleration.

## 5 Conclusion

In solving some key challenges, this work enables a new tool for the 3-d reconstruction task: refining structure and camera parameters jointly, using a photometric error that is robust to local lighting variations. The framework was evaluated on 15 sets of 150-700 images, with a variety of subject matter. The result is a significant, broad increase in the metric accuracy of reconstruction (up to scale), over a baseline that is representative of the current approach used on this problem: feature-based SfM followed by photometric MVS. Our ablation study provides valuable insight into exactly which aspects of this new approach deliver the most benefit, highlighting the gain in accuracy due specifically to such a refinement.

We have not presented a full system, nor optimized peripheral aspects of the framework, such as landmark selection or visibilities, source frame indices, the robust kernel, landmark weights, or patch sample spacing. We rely on other methods for initialization, which may fail. Improvements are possible in all these areas. Also, we do not propose a replacement to traditional MVS; such systems are complementary, and can be applied after a photometric refinement (which could then use far fewer landmarks), as we show, taking advantage of improved camera pose and intrinsic estimates. We note, however, that our framework could also be incorporated into an MVS method (or surface priors could be added to our method), where all camera parameters are fixed, as well as VO methods, where intrinsic parameters are fixed. Indeed, two widely used MVS frameworks, PMVS [10] and COLMAP-MVS [26], both use NCC, but neither currently use a second order optimizer or analytic gradients.

<sup>8</sup>colmap automatic\_reconstructor -single\_camera -quality low. The density of landmarks is lower (though average precision can be higher), so results are not directly comparable to other rows.

<sup>9</sup>The two for loops in Algorithm 1 are easily parallelized, *e.g.* using OpenMP.

## References

- [1] Sameer Agarwal, Keir Mierle, and Others. Ceres solver. <http://ceres-solver.org>.
- [2] Hatem Alismail, Brett Browning, and Simon Lucey. Photometric bundle adjustment for vision-based slam. In *Proceedings of the Asian Conference on Computer Vision*, pages 324–341. Springer, 2016.
- [3] Hatem Alismail, Michael Kaess, Brett Browning, and Simon Lucey. Direct visual odometry in low light using binary descriptors. *IEEE Robotics and Automation Letters*, 2(2):444–451, 2016.
- [4] Serge Belongie. Rodrigues’ rotation formula. From MathWorld—A Wolfram Web Resource, created by Eric W. Weisstein. <http://mathworld.wolfram.com/RodriguesRotationFormula.html>.
- [5] Michael J. Black and Anand Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *International Journal of Computer Vision*, 19(1):57–91, 1996.
- [6] Amaël Delaunoy and Marc Pollefeys. Photometric bundle adjustment for dense multi-view 3d modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1486–1493, 2014.
- [7] Pierre Drap and Julien Lefèvre. An exact formula for calculating inverse radial lens distortions. *Sensors*, 16(6):807, 2016.
- [8] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4, 2017.
- [9] Yasutaka Furukawa and Jean Ponce. Accurate camera calibration from multi-view stereo and bundle adjustment. *International Journal of Computer Vision*, 84(3):257–268, 2009.
- [10] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8):1362–1376, 2010.
- [11] Stuart Geman and Donald E. McClure. Bayesian image analysis: An application to single photon emission tomography. *Amer. Statist. Assoc.*, pages 12–18, 1985.
- [12] Bastian Goldlücke, Mathieu Aubry, Kalin Kolev, and Daniel Cremers. A super-resolution framework for high-accuracy multiview reconstruction. *International Journal of Computer Vision*, 106(2):172–191, 2014.
- [13] Gene H Golub and Victor Pereyra. The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. *SIAM Journal on Numerical Analysis*, 10(2):413–432, 1973.
- [14] Martin Habbecke and Leif Kobbelt. Iterative multi-view plane fitting. In *Int. Fall Workshop of Vision, Modeling, and Visualization*, pages 73–80, 2006.

- [15] Paul W. Holland and Roy E. Welsch. Robust regression using iteratively reweighted least-squares. *Communications in Statistics-theory and Methods*, 6(9):813–827, 1977.
- [16] Je Hyeong Hong, Christopher Zach, Andrew Fitzgibbon, and Roberto Cipolla. Projective bundle adjustment from arbitrary initialization using the variable projection method. In *Proceedings of the European Conference on Computer Vision*, pages 477–493. Springer, 2016.
- [17] Olaf Kähler and Joachim Denzler. Tracking and reconstruction in a combined optimization approach. 34(2):387–401, 2011.
- [18] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics*, 32(3):1–13, 2013.
- [19] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4), 2017. <https://www.tanksandtemples.org>.
- [20] Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics*, 2(2):164–168, 1944.
- [21] Bruce D Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.
- [22] Onur Özyeşil, Vladislav Voroninski, Ronen Basri, and Amit Singer. A survey of structure from motion. *Acta Numerica*, 26:305–364, 2017.
- [23] Seonwook Park, Thomas Schöps, and Marc Pollefeys. Illumination change robustness in direct visual slam. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4523–4530. IEEE, 2017.
- [24] Johannes L. Schönberger. Colmap. <https://colmap.github.io>.
- [25] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.
- [26] Johannes L. Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *Proceedings of the European Conference on Computer Vision*, pages 501–518. Springer, 2016.
- [27] Noah Snavely, Steven M Seitz, and Richard Szeliski. Modeling the world from internet photo collections. *International Journal of Computer Vision*, 80(2):189–210, 2008.
- [28] D. Strelow, Q. Wang, L. Si, and A. Eriksson. General, nested, and constrained wiberg minimization. 38(9):1803–1815, 2016.
- [29] O Tingleff, K Madsen, and HB Nielsen. Methods for non-linear least squares problems. *Lecture Note in Computer Science 02611 Optimization and Data Fitting*, 2004.
- [30] Oliver J. Woodford. Using normalized cross correlation in least squares optimizations. 2018. URL <http://arxiv.org/abs/1810.04320>.