Color Map Optimization for 3D Reconstruction with Consumer Depth Cameras

Qian-Yi Zhou[*] Vladlen Koltun[†]

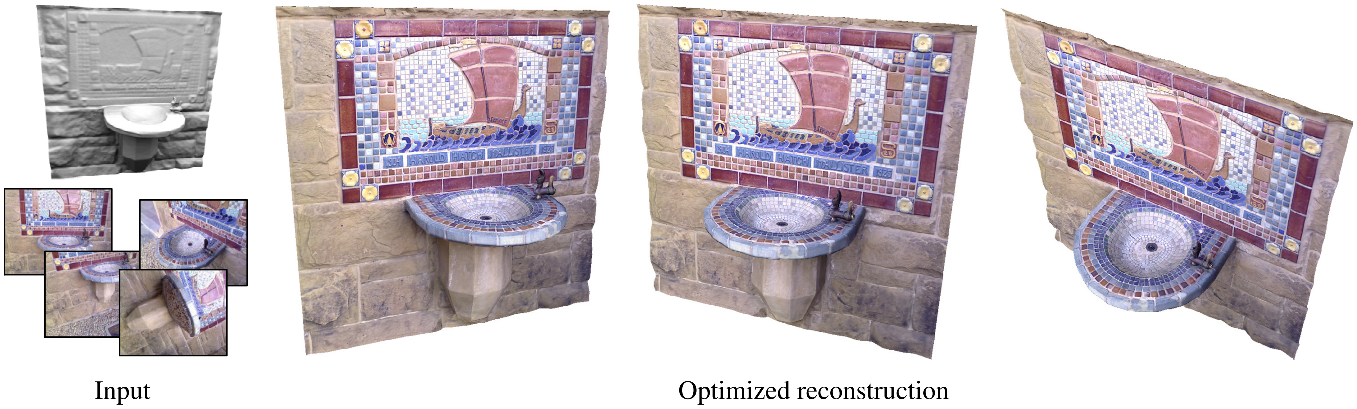


Figure 1: *Given a geometric model and corresponding color images produced by a consumer-grade RGB-D camera (left), our approach optimizes a photometrically consistent mapping of the images to the model.*

Abstract

We present a global optimization approach for mapping color images onto geometric reconstructions. Range and color videos produced by consumer-grade RGB-D cameras suffer from noise and optical distortions, which impede accurate mapping of the acquired color data to the reconstructed geometry. Our approach addresses these sources of error by optimizing camera poses in tandem with non-rigid correction functions for all images. All parameters are optimized jointly to maximize the photometric consistency of the reconstructed mapping. We show that this optimization can be performed efficiently by an alternating optimization algorithm that interleaves analytical updates of the color map with decoupled parameter updates for all images. Experimental results demonstrate that our approach substantially improves color mapping fidelity.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling

Keywords: 3d reconstruction, texture mapping, range imaging, consumer depth cameras, optimization

Links: DL PDF

1 Introduction

Consumer depth cameras are now widely available. Tens of millions of such cameras have been shipped and miniaturized versions are being developed for integration into laptops, tablets, and smartphones. As a result, millions of people can now create high-fidelity geometric models of real-world objects and scenes [Newcombe et al. 2011; Chen et al. 2013; Zhou and Koltun 2013; Zhou et al. 2013].

However, capturing an object's geometry is not sufficient for reproducing its appearance. A visually faithful reconstruction must also incorporate the apparent color of every point on the object. In this respect the results demonstrated by recent reconstruction systems are less impressive. To color geometric models produced using consumer depth cameras, existing systems use a volumetric blending approach that integrates color samples over a voxel grid [Izadi et al. 2011; Nießner et al. 2013; Whelan et al. 2013; Bylow et al. 2013; Sturm et al. 2013; Endres et al. 2014]. This produces color maps that convey the object's general appearance, but suffer from blurring, ghosting, and other visual artifacts that are apparent at close range.

There are several factors that diminish the quality of reconstructed color maps. First, the geometric model for which the color map is constructed is produced from noisy data and is inaccurate. Second, the camera poses that are used to map the input images onto the model are estimated from the same noisy data and are likewise imprecise. Third, the shutters of depth and color cameras in consumer devices are not perfectly synchronized, thus for handheld scans the color camera is not in rigid correspondence with the depth camera: this further increases the misalignment of the projected images. Fourth, color images produced by consumer RGB-D cameras suffer from optical distortions that are not accounted for by the pinhole camera model.

In this paper, we describe an approach for optimizing the mapping of color images produced by a handheld RGB-D camera to a corresponding geometric reconstruction. Our approach addresses the aforementioned difficulties in a coherent optimization framework. To correct imprecise camera localization, we jointly optimize the

[*] Stanford University
[†] Adobe Research

poses of all color frames. To correct the complex distortions introduced by inaccuracies in the geometric model and the optical path, we optimize a non-rigid correction function for each image. The camera poses and the non-rigid corrections for all images are optimized in tandem to maximize a single joint objective: the photometric consistency of the mapping.

We describe an alternating optimization algorithm that effectively decouples the variable updates for different images. Global coordination is achieved through simple analytical updates of the color map. As a result, despite the large number of optimized parameters and the challenging nonlinear objective, our approach produces globally optimized mappings in minutes.

We have evaluated the presented approach on a large number of handheld scans of indoor and outdoor objects. Experimental results demonstrate that our approach substantially increases the fidelity of reconstructed color maps.

2 Related Work

Creating a color map for a geometric model given multiple images of the depicted object is a classical problem in computer graphics. A commonly used approach is to compute a color for each point on the model by averaging the image colors at corresponding image locations. In many works, the registration between the images and the model is assumed to be provided, by means of manually selected point correspondences [Ofek et al. 1997; Pighin et al. 1998; Neugebauer and Klein 1999; Rocchini et al. 1999; Stamos and Allen 2000; Yamauchi et al. 2005; Franken et al. 2005], photogrammetry targets [Baumberg 2002], or high-end imaging setups [Levoy et al. 2000]. In our case, the images can be registered to the model by the geometric reconstruction pipeline, but this geometric registration is not sufficiently precise for producing high-fidelity color maps.

A number of works explore camera pose optimization to maximize color agreement. If registered pairs of range and color images are given, color consistency can be optimized as part of the geometric alignment process [Johnson and Kang 1999; Pulli and Shapiro 2000; Bernardini et al. 2001; Pulli et al. 2005]. In our case, neither the range data nor the registration between the range and color images is sufficiently precise for this approach to produce color maps of the quality we seek. A number of researchers have investigated optimization approaches that align features in the color images to features in the geometric model [Lensch et al. 2001; Stamos and Allen 2002; Corsini et al. 2009; Liu and Stamos 2012]. In our setting, the geometric model is not sufficiently precise for these approaches to suffice. Ikeuchi et al. [2007] align color images to the geometric model by registering the images to laser reflectance data, which is produced as a byproduct of laser range finding. Corsini et al. [2013] describe a graph-based pose refinement algorithm that maximizes the mutual information between projected images. We also perform global optimization, but our objective and optimization approach are different and naturally accommodate non-rigid deformation functions that correct residual misalignment for all images.

A small number of recent research efforts have considered the use of non-rigid deformation to resolve complex misalignments due to imprecise geometry and other factors. Aganj et al. [2009] search for corresponding SIFT keypoints in the input images and analytically compute displacement vectors that align matched keypoints; the displacement is then interpolated using thin-plate splines. Our approach does not rely on keypoint extraction and matching, which are far from perfect in practice; rather, we optimize a dense photo-consistency term that takes full image information into account and is naturally coupled with rigid camera localization. Gal et al. [2010] assign each face in the geometric model to a single input image and optimize a per-face translational shift within the image to minimize seams. Face-to-image assignments and quantized shifts are computed using combinatorial optimization. This approach does not attempt to perform camera localization. It is also computationally expensive: the running time of the algorithm is quite high even for a moderate number of triangles in the model and a small number of color images. In contrast, we show that continuous non-rigid distortion functions for all images can be computed jointly with optimized camera poses using a global optimization approach that can handle models with hundreds of thousands of triangles in minutes. Dellepiane et al. [2012] compute optical flow between pairs of images and blend the computed flow fields. We show that distortion functions for all images can be optimized together, by a single optimization procedure that minimizes a clear global objective.

In contrast to all of the above approaches, we optimize the camera poses in tandem with the non-rigid correction functions. All parameters are computed by a coherent global optimization algorithm that minimizes a single joint objective: the photometric consistency of the computed mapping. Our experiments demonstrate that joint optimization of camera poses and non-rigid distortion parameters produces superior results to optimization of either of these components alone.

Once the mapping of the color images onto the geometric model is computed, a number of techniques can be used to integrate samples from different images. A common approach is to average the samples and sophisticated averaging schemes have been developed [Callieri et al. 2008]. Alternatively, a color assignment can be computed by solving a Poisson equation [Chuang et al. 2009; Li et al. 2013] or a discrete labeling problem [Lempitsky and Ivanov 2007; Sinha et al. 2008]. Our work is complementary to these ideas: once a mapping of the color images to the geometry is optimized, any of these integration techniques can be used.

3 Overview

Input. Our input is a stream of depth images and a stream of high-resolution color images. We use an Asus Xtion Pro Live camera, which streams VGA resolution depth images at 30 fps and SXGA (1280x1024) color images at 10 fps. (The camera can stream VGA resolution color images at 30 fps, but we value resolution over frame rate and use the highest resolution color images, which are streamed at a lower frame rate.) The images in the two streams are time-stamped by a common clock. The shutters are not in sync, but the time stamps can be used to match color images to the closest depth images. The color images are captured with fixed exposure and white balancing.

Geometric reconstruction. To create the geometric model we use KinectFusion [Newcombe et al. 2011; Izadi et al. 2011]. This produces an initial mesh $\mathbf{M}^0$ along with estimated camera poses that approximately register each depth image to the mesh. The resolution of this initial mesh is determined by the resolution of the voxel grid used by the KinectFusion pipeline. We subdivide this mesh multiple times to obtain the final mesh $\mathbf{M}$ [Peters and Reif 1997]. Each triangle is recursively subdivided within its plane, thus the subdivision does not alter the actual geometry of the model. Let $\mathbf{P}$ denote the set of vertices of $\mathbf{M}$. Our goal is to compute a color for each vertex $\mathbf{p} \in \mathbf{P}$. These colors are linearly interpolated within the triangles of $\mathbf{M}$.

Key frames. Since the camera is handheld, many of the input color images suffer from strong motion blur. To maximize the quality of the computed color map, our pipeline automatically selects a subset of the input images. Specifically, we quantify the blurriness of each image using the metric of Crete et al. [2007] and select key frames greedily as follows. Given a set of already selected key frames, we add the frame that has the lowest blurriness in the time segment (t_-, t_+) after the last selected key frame. Parameters t_- and t_+ determine the upper and lower bounds on key frame density. In our implementation, $(t_-, t_+) = (1, 5)$ seconds. The set of key frames is denoted by $\{I_i\}$. This is the set of images used in the subsequent stages of the pipeline.

For each key frame I_i, we render the mesh $\mathbf{M}$ onto the image plane of I_i using the camera pose of the closest depth image and the default intrinsic parameters of the camera. A visibility test is applied by comparing the depth value of each vertex and the corresponding value in the depth buffer produced for the rendering. This is used to identify a set of vertices of $\mathbf{M}$ that are potentially visible in key frame I_i. This set is pruned by filtering out vertices whose projections are within distance δ from image boundaries or depth discontinuities. (In our implementation, δ is set to 9 pixels.) The remaining vertices are denoted by $\mathbf{P}_i \subset \mathbf{P}$.

Optimization. This is the heart of our approach. Given color images $\{I_i\}$ and corresponding vertex subsets $\mathbf{P}_i$, we optimize a photometric consistency objective that quantifies, for each vertex $\mathbf{p} \in \mathbf{P}$, the color consistency of all corresponding image locations. The objective and the optimization algorithm are the subject of Sections 4 and 5. For the sake of exposition, we begin in Section 4 by treating the problem of optimizing the camera poses for all images. We then show in Section 5 how the presented approach is extended to optimize the camera poses in concert with non-rigid correction functions for all images.

4 Camera Pose Optimization

In this section we present the approach in a restricted setting in which only the camera poses are optimized. This reduces notational clutter when the objective is introduced and accelerates the presentation of optimization strategies. The photometric consistency objective is introduced in Section 4.1. Section 4.2 describes a natural application of the Gauss-Newton method to this objective. Section 4.3 then presents an alternating optimization algorithm.

4.1 Objective

Our input is the set of images $\{I_i\}$ and the associated vertex subsets $\{\mathbf{P}_i\}$. For each image I_i, we want to optimize an extrinsic matrix $\mathbf{T}_i$ that maps vertices in $\mathbf{P}_i$ from the global frame of the geometric model $\mathbf{M}$ to the local frame of the image. We use homogeneous coordinates, thus $\mathbf{P}_i \subset \mathbb{P}^3$ and $\mathbf{T}_i$ is a 4×4 matrix.

Our objective is to maximize for each point $\mathbf{p} \in \mathbf{P}$ the agreement of the colors of $\mathbf{p}$ in all images associated with $\mathbf{p}$. Specifically, consider the set of images $I_\mathbf{p} = \{I_i : \mathbf{p} \in \mathbf{P}_i\}$. This is the set of images associated with $\mathbf{p}$. Let $\Gamma_i(\mathbf{p}, \mathbf{T}_i)$ be the color at the image coordinates of the projection of $\mathbf{p}$ onto $I_i \in I_\mathbf{p}$, given an extrinsic matrix $\mathbf{T}_i$. We want to maximize the agreement within $\{\Gamma_i(\mathbf{p}, \mathbf{T}_i)\}_{I_i \in I_\mathbf{p}}$, for each $\mathbf{p}$. To this end, we introduce an auxiliary variable $C(\mathbf{p})$, which serves as a proxy for the color of $\mathbf{p}$. For simplicity, we use greyscale images for the optimization. Thus $C(\mathbf{p})$ and $\Gamma_i(\mathbf{p}, \mathbf{T}_i)$ are scalars. Let $\mathbf{C} = \{C(\mathbf{p})\}$ and $\mathbf{T} = \{\mathbf{T}_i\}$. Our goal is to optimize the set of camera transforms $\mathbf{T}$, with $\mathbf{C}$ serving as auxiliary variables. We minimize the following objective:

$$E(\mathbf{C}, \mathbf{T}) = \sum_i \sum_{\mathbf{p} \in \mathbf{P}_i} \big(C(\mathbf{p}) - \Gamma_i(\mathbf{p}, \mathbf{T}_i)\big)^2. \tag{1}$$

To perform the optimization, we need to be more specific about how $\Gamma_i(\mathbf{p}, \mathbf{T}_i)$ is computed. $\Gamma_i(\mathbf{p}, \mathbf{T}_i)$ is produced by a composition of a rigid transformation, a projection, and a color evaluation, which can be expressed as $\Gamma_i(\mathbf{u}(\mathbf{g}(\mathbf{p}, \mathbf{T}_i)))$. Here $\mathbf{g}$ is the rigid transformation:

$$\mathbf{g}(\mathbf{p}, \mathbf{T}_i) = \mathbf{T}_i \mathbf{p}. \tag{2}$$

$\mathbf{u}$ is the projection onto the image plane of I_i:

$$\mathbf{u}(g_x, g_y, g_z, g_w) = \left(\frac{g_x f_x}{g_z} + c_x, \frac{g_y f_y}{g_z} + c_y\right)^\top, \tag{3}$$

where f_x and f_y are the focal lengths and $(c_x, c_y)^\top$ is the principal point. Finally, $\Gamma_i(u_x, u_y)$ is the color evaluation that returns the bilinearly interpolated greyscale intensity for coordinates (u_x, u_y) in image I_i.

Our objective can now be written as

$$E(\mathbf{C}, \mathbf{T}) = \sum_i \sum_{\mathbf{p} \in \mathbf{P}_i} r_{i,\mathbf{p}}^2, \tag{4}$$

where $r_{i,\mathbf{p}}$ is the residual

$$r_{i,\mathbf{p}} = C(\mathbf{p}) - \Gamma_i(\mathbf{u}(\mathbf{g}(\mathbf{p}, \mathbf{T}_i))). \tag{5}$$

4.2 Gauss-Newton Method

$E(\mathbf{C}, \mathbf{T})$ is a nonlinear least-squares objective and can therefore be minimized using the Gauss-Newton method. Let $\mathbf{x}$ be the vector of variables that includes all the parameters of $\mathbf{C}$ and $\mathbf{T}$. The optimization is initialized by a parameter vector $\mathbf{x}^0 = [\mathbf{C}^0, \mathbf{T}^0]$. For each i, $\mathbf{T}_i^0$ is provided by the camera pose of the closest depth image to I_i. For each $\mathbf{p}$, $C^0(\mathbf{p})$ is set to the average of $\{\Gamma_i(\mathbf{p}, \mathbf{T}_i^0)\}_{I_i \in I_\mathbf{p}}$. Each iteration of the algorithm updates $\mathbf{x}$ as follows:

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \Delta\mathbf{x}, \tag{6}$$

where $\Delta\mathbf{x}$ is the solution to the following linear system:

$$\mathbf{J_r}^\top \mathbf{J_r} \Delta\mathbf{x} = -\mathbf{J_r}^\top \mathbf{r}. \tag{7}$$

Here $\mathbf{r} = \mathbf{r}(\mathbf{x})$ is the residual vector and $\mathbf{J_r} = \mathbf{J_r}(\mathbf{x})$ is the Jacobian of $\mathbf{r}$. Both are evaluated at $\mathbf{x}^k$:

$$\mathbf{r} = \left[r_{i,\mathbf{p}}(\mathbf{x})|_{\mathbf{x}=\mathbf{x}^k}\right]_{(i,\mathbf{p})}, \tag{8}$$

$$\mathbf{J_r} = \left[\nabla r_{i,\mathbf{p}}(\mathbf{x})|_{\mathbf{x}=\mathbf{x}^k}\right]_{(i,\mathbf{p})}. \tag{9}$$

In each iteration, $\mathbf{r}$ can be computed using equation (5). The partial derivatives of $r_{i,\mathbf{p}}$ with respect to $\mathbf{C}$ and $\mathbf{T}_j|_{j \neq i}$ are trivial. To compute the partial derivatives of $r_{i,\mathbf{p}}$ with respect to $\mathbf{T}_i$, we locally linearize $\mathbf{T}_i$ around $\mathbf{T}_i^k$. Specifically, we parameterize $\mathbf{T}_i$ by a 6-vector $\xi_i = (\alpha_i, \beta_i, \gamma_i, a_i, b_i, c_i)^\top$ that represents an incremental transformation relative to $\mathbf{T}_i^k$. Here $(a_i, b_i, c_i)^\top$ is translation and $(\alpha_i, \beta_i, \gamma_i)^\top$ can be interpreted as angular velocity. $\mathbf{T}_i$ is thus approximated by a linear function of ξ_i:

$$\mathbf{T}_i \approx \begin{pmatrix} 1 & -\gamma_i & \beta_i & a_i \\ \gamma_i & 1 & -\alpha_i & b_i \\ -\beta_i & \alpha_i & 1 & c_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \mathbf{T}_i^k. \tag{10}$$

With this parametrization, $\Delta\mathbf{x}$ is a vector that collates $\{C(\mathbf{p})\}$ and $\{\xi_i\}$. To compute the partial derivative of $r_{i,\mathbf{p}}$ with respect to $\mathbf{T}_i$, we use equation (5) and apply the chain rule:

$$\nabla r_{i,\mathbf{p}}(\xi_i)|_{\mathbf{x}=\mathbf{x}^k} = -\frac{\partial}{\partial \xi_i}(\Gamma_i \circ \mathbf{u} \circ \mathbf{g})\big|_{\mathbf{x}=\mathbf{x}^k} \tag{11}$$

$$= -\nabla\Gamma_i(\mathbf{u})\mathbf{J_u}(\mathbf{g})\mathbf{J_g}(\xi_i)|_{\mathbf{x}=\mathbf{x}^k}, \tag{12}$$

where $\nabla\Gamma_i$ is the gradient of Γ_i, computed by applying a normalized Scharr kernel over the greyscale image, $\mathbf{J}_{\mathbf{u}}(\mathbf{g})$ is the Jacobian of $\mathbf{u}$, derived from equation (3), and $\mathbf{J}_{\mathbf{g}}(\xi_i)$ is the Jacobian of $\mathbf{g}$ with respect to ξ_i, derived from equations (2) and (10).

In each iteration, we evaluate the residual vector $\mathbf{r}$ and the Jacobian $\mathbf{J}_{\mathbf{r}}$ at $\mathbf{x}^k$, and solve the linear system (7) using sparse Cholesky factorization. Although the linear system has $m + 6n$ variables, where m is the number of vertices and n the number of key frames, the matrix $\mathbf{J}_{\mathbf{r}}^\top \mathbf{J}_{\mathbf{r}}$ is sparse and symmetric positive definite. Thus the linear system can be solved in a small number of hours for problems with hundreds of thousands of vertices and dozens of key frames. (Precise timings are provided in Section 6.) The solution $\Delta\mathbf{x}$ is used to update $\mathbf{x}$ according to equation (6). Specifically, camera extrinsics are updated using equation (10) and mapped back into the $SE(3)$ group. In the next iteration, we re-parameterize $\mathbf{T}_i$ around $\mathbf{T}_i^{k+1}$ and repeat.

4.3 Alternating Optimization

We now present an alternating optimization scheme for minimizing the objective. This approach has extremely favorable scalability characteristics. The basic idea is to alternate between optimizing $\mathbf{C}$ and optimizing $\mathbf{T}$. When $\mathbf{C}$ is optimized $\mathbf{T}$ is kept fixed and vice versa. Thus in each iteration of this scheme, all variables $\mathbf{C}$ and $\mathbf{T}$ are optimized, but this optimization is performed in two stages. In each stage, some variables are fixed but the same global objective is optimized. Thus the algorithm is guaranteed to converge.

When $\mathbf{T}$ is fixed, the nonlinear least-squares problem (4) turns into a *linear* least-squares problem that has a closed form solution:

$$C(\mathbf{p}) = \frac{1}{n_{\mathbf{p}}} \sum_{I_i \in I_{\mathbf{p}}} \Gamma_i(\mathbf{p}, \mathbf{T}_i), \tag{13}$$

where $n_{\mathbf{p}} = |I_{\mathbf{p}}|$ is the number of images associated with $\mathbf{p}$. Thus we simply need to compute the average intensity of the projections of $\mathbf{p}$ onto the images associated with $\mathbf{p}$.

When $\mathbf{C}$ is fixed, the objective (4) decomposes into *independent* objectives for each $\mathbf{T}_i$:

$$E_i(\mathbf{T}) = \sum_{\mathbf{p} \in \mathbf{P}_i} r_{i,\mathbf{p}}^2. \tag{14}$$

Each term $E_i(\mathbf{T})$ involves only the six variables ξ_i. In each iteration, we update these variables using a Gauss-Newton step that has the general form of (7) but only 6 variables. The Gauss-Newton update for each $\mathbf{T}_i$ is thus independent and can be computed in parallel. Overall, the scheme is extremely efficient: instead of solving a linear system with $m + 6n$ variables, each iteration reduces to solving n linear systems of 6 variables.

5 Non-Rigid Correction

We now extend the approach to incorporate a non-rigid correction for each image that can rectify complex distortions due to imprecise geometry and optical aberrations.

5.1 Objective

The non-rigid correction is represented as a deformation function $\mathbf{F}_i$ over the image plane of I_i, for each image. We explicitly represent the deformation over a control lattice $\mathbf{V}_i$ and interpolate it over the continuous domain. Let the set of control vertices in $\mathbf{V}_i$ be $\{\mathbf{v}_{i,l}\}$. Let the correction applied by $\mathbf{F}_i$ to $\mathbf{v}_{i,l}$ be $\mathbf{f}_{i,l}$. Thus

$$\mathbf{F}_i(\mathbf{v}_{i,l}) = \mathbf{v}_{i,l} + \mathbf{f}_{i,l}.$$

The correction $\mathbf{f}_{i,l}$ is simply a two-dimensional vector. The deformation $\mathbf{F}_i$ is extended to all points $\mathbf{u}$ in the image plane as

$$\mathbf{F}_i(\mathbf{u}) = \mathbf{u} + \sum_l \theta_l(\mathbf{u})\mathbf{f}_{i,l}. \tag{15}$$

Here θ_l are the basis functions for bilinear interpolation. Thus for each point $\mathbf{u}$ in the image plane, the correction $\mathbf{F}_i(\mathbf{u})$ is a linear combination of corrections $\{\mathbf{F}_i(\mathbf{v}_{i,l})\}_{\mathbf{v}_{i,l} \in \mathbf{V}_i}$. Only a small number of coefficients $\theta_l(\mathbf{u})$ are non-zero at any point $\mathbf{u}$. In our implementation, $\mathbf{V}_i$ is an orthogonal grid with 20×16 cells (and 21×17 control vertices).

Let $\mathbf{F} = \{\mathbf{f}_{i,l}\}_{(i,l)}$ be the set of parameters of the non-rigid correction functions for all images. The photometric consistency objective can now be rephrased to incorporate the effect of non-rigid corrections:

$$E_c(\mathbf{C}, \mathbf{T}, \mathbf{F}) = \sum_i \sum_{\mathbf{p} \in \mathbf{P}_i} r_{i,\mathbf{p}}^2, \tag{16}$$

where the residual is

$$r_{i,\mathbf{p}} = C(\mathbf{p}) - \Gamma_i(\mathbf{F}_i(\mathbf{u}(\mathbf{g}(\mathbf{p}, \mathbf{T}_i)))). \tag{17}$$

To prevent the correction functions from drifting we add a simple L^2 regularizer on the magnitude of the offsets $\mathbf{f}_{i,l}$:

$$E_r(\mathbf{F}) = \sum_i \sum_l \mathbf{f}_{i,l}^\top \mathbf{f}_{i,l}. \tag{18}$$

Our complete objective is now

$$E(\mathbf{C}, \mathbf{T}, \mathbf{F}) = E_c(\mathbf{C}, \mathbf{T}, \mathbf{F}) + \lambda E_r(\mathbf{F}), \tag{19}$$

where λ is a coefficient that balances the strength of the two terms. λ is related to the density of projected points $\mathbf{u}(\mathbf{g}(\mathbf{p}, \mathbf{T}_i))$ in the grid cells of $\mathbf{V}$, since the number of regularization term summands grows as a function of grid vertices and the number of data term summands grows as a function of projected mesh vertices. We use $\lambda = 0.1$ in all our experiments.

5.2 Optimization

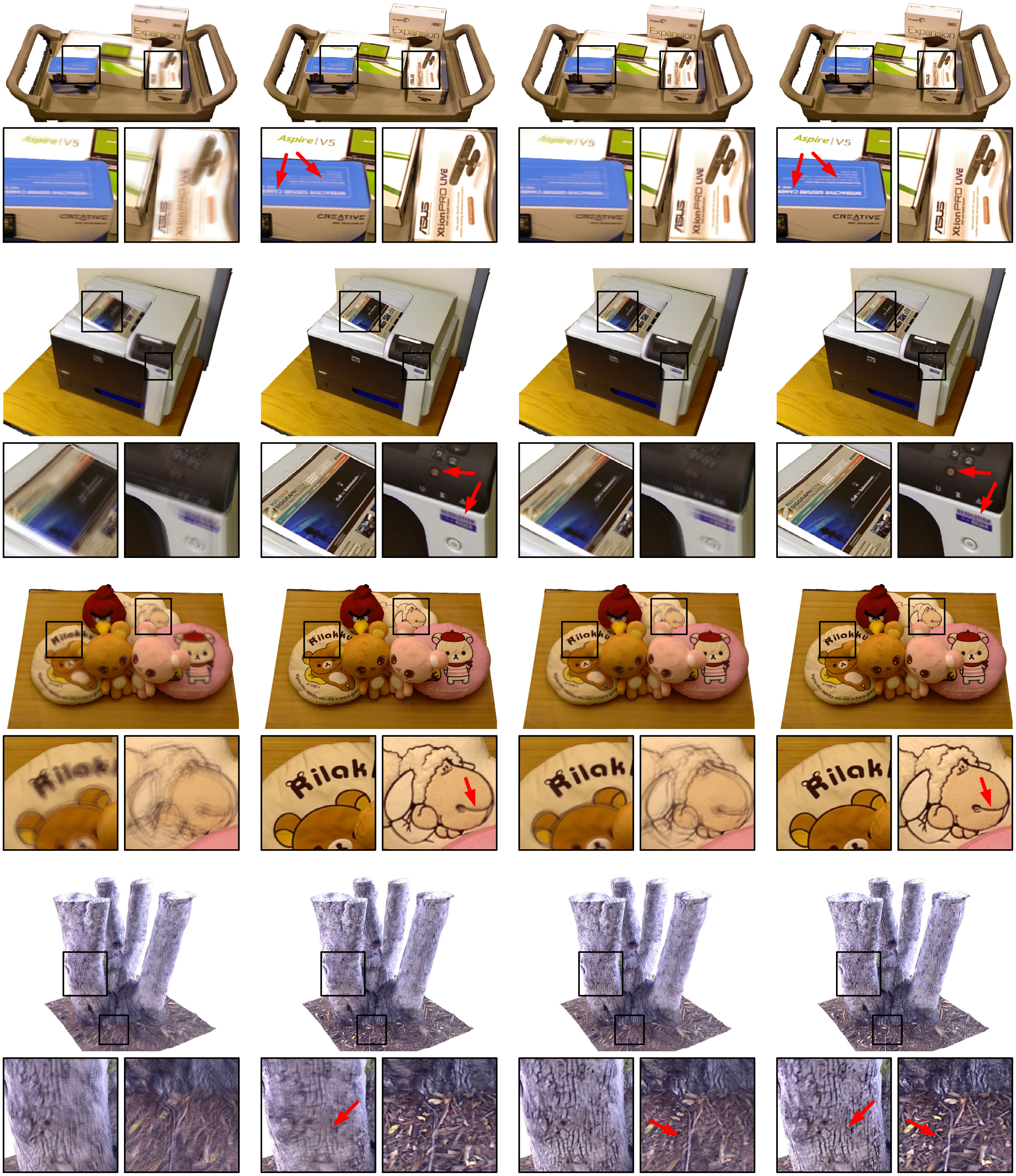
The key advantage of the alternating optimization algorithm is its scalability. The objective (19) involves $m + 720n$ variables and a straightforward application of the Gauss-Newton method would be extremely computationally expensive. In contrast, the alternating optimization strategy reduces to solutions of independent linear systems with 720 variables.

As in Section 4.3, we proceed iteratively. In the kth iteration, we first fix $\mathbf{T}$ and $\mathbf{F}$ and optimize $\mathbf{C}$. This can be done in closed form as in equation (13). We then fix $\mathbf{C}$ and optimize $\mathbf{T}$ and $\mathbf{F}$. The objective (19) decomposes into n independent objectives over the individual images. We perform a Gauss-Newton update for each image. The Jacobian of the regularization term is straightforward. The non-zero entries of the Jacobian for the photometric consistency term are as follows:

$$\begin{aligned} \nabla r_{i,\mathbf{p}}(\xi_i)|_{\mathbf{x}=\mathbf{x}^k} &= -\nabla\Gamma_i(\mathbf{F})\mathbf{J}_{\mathbf{F}_i}(\mathbf{u})\mathbf{J}_{\mathbf{u}}(\mathbf{g})\mathbf{J}_{\mathbf{g}}(\xi_i)|_{\mathbf{x}=\mathbf{x}^k}, \\ \nabla r_{i,\mathbf{p}}(\mathbf{f}_{i,l})|_{\mathbf{x}=\mathbf{x}^k} &= -\theta_l(\mathbf{u})\nabla\Gamma_i(\mathbf{F})|_{\mathbf{x}=\mathbf{x}^k}, \end{aligned}$$

where $\mathbf{J}_{\mathbf{F}_i}(\mathbf{u})$ is the Jacobian of $\mathbf{F}_i$ with respect to $\mathbf{u}$:

$$\mathbf{J}_{\mathbf{F}_i}(\mathbf{u}) = \mathbf{I} + \sum_l \mathbf{f}_{i,l} \nabla\theta_l(\mathbf{u}). \tag{20}$$



(a) No optimization

(b) Camera pose only

(c) Non-rigid correction only

(d) Complete objective

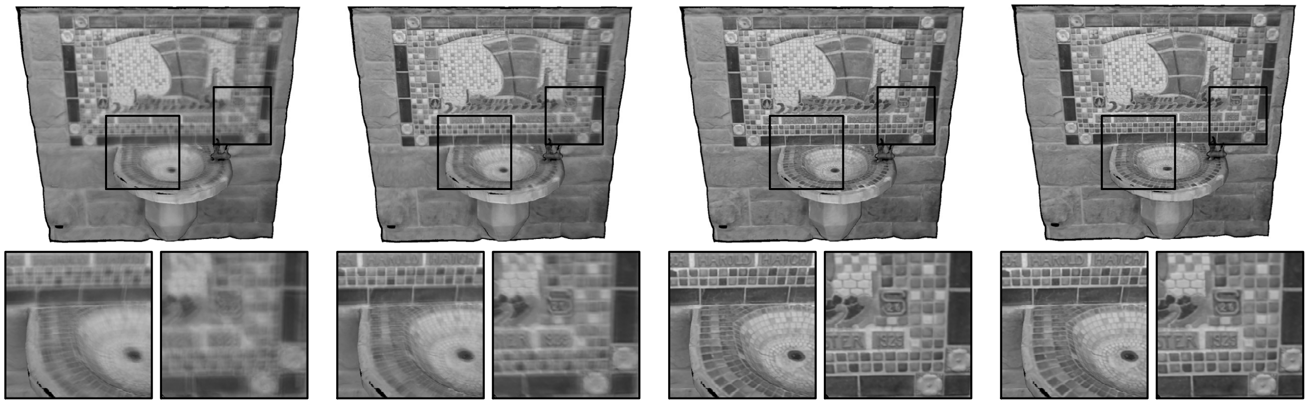
Figure 2: *Effect of camera pose and non-rigid correction optimization. (a) Initial alignment, (b) result of optimizing camera poses without non-rigid correction, (c) result of optimizing non-rigid corrections with fixed camera poses, (d) result of joint optimization of camera poses and non-rigid corrections. Corresponding quantitative results are provided in Table 2.*

Model	Residual error using the Gauss-Newton method				Time per iteration	Residual error using alternating optimization				Time per iteration
	Initial	10 iter.	50 iter.	200 iter.		Initial	10 iter.	50 iter.	200 iter.	
Figure 1	0.081	0.064	0.050	0.049	18.30s	0.081	0.068	0.050	0.049	0.64s
Figure 2, row 1	0.092	0.046	0.041	0.041	26.73s	0.092	0.054	0.042	0.041	0.66s
Figure 2, row 2	0.055	0.039	0.033	0.033	39.34s	0.055	0.040	0.033	0.032	1.02s
Figure 2, row 3	0.062	0.045	0.036	0.035	33.83s	0.062	0.048	0.036	0.035	0.79s
Figure 2, row 4	0.095	0.089	0.079	0.068	16.07s	0.095	0.090	0.081	0.072	0.59s

Table 1: *Normalized residual error and average time per iteration for camera pose optimization, using the Gauss-Newton method (left) and alternating optimization (right). Alternating optimization achieves similar accuracy but is substantially faster.*

Model	# of points	# of key frames	Initial error	Camera pose only		Non-rigid correction only		Complete objective	
				Error	Time	Error	Time	Error	Time
Figure 1	536,872	33	0.081	0.049	128s	0.064	209s	0.045	296s
Figure 2, row 1	504,963	53	0.092	0.041	132s	0.050	208s	0.037	293s
Figure 2, row 2	874,130	43	0.055	0.033	204s	0.041	334s	0.030	470s
Figure 2, row 3	489,839	50	0.062	0.035	158s	0.050	249s	0.033	355s
Figure 2, row 4	892,378	32	0.095	0.072	118s	0.080	165s	0.061	235s

Table 2: *Effect of camera pose and non-rigid correction optimization. Joint optimization of camera poses and non-rigid corrections improves on optimizing either of the two components alone.*



Initialization 10 iterations 50 iterations 200 iterations

Figure 3: *Progress of alternating optimization on the model shown in Figure 1. The complete objective is being optimized, including non-rigid correction functions. The images visualize the values of the proxy variables $C(\mathbf{p})$ for vertices $\mathbf{p}$ in the model. Some of the vertices are shown black and do not have corresponding variables, since they were automatically masked out in all key frames due to proximity to image boundaries or depth discontinuities. These vertices do not bias the optimization but are given a color in the final reconstructed color map, as shown in Figure 1 and described in Section 6.*

6 Results

We begin by evaluating the relative performance of the Gauss-Newton method and the alternating optimization algorithm. This evaluation is performed on camera pose optimization, so that the experiment can be conducted in a reasonably short time frame (up to about two and a half hours per model for the Gauss-Newton method). All experiments were performed on a workstation with an Intel i7 3.2GHz CPU and 24GB of RAM. The results are reported in Table 1. The CHOLMOD package was used for solving linear systems. The two algorithms achieve similar gains on the objective, but the alternating optimization algorithm is much faster. Alternating optimization for 200 iterations is used in all subsequent experiments.

Next we evaluate the contribution of camera pose optimization and non-rigid correction. Qualitative results are shown in Figure 2 and quantitative results are reported in Table 2. Joint optimization of camera poses and non-rigid correction functions produces superior results both qualitatively and quantitatively. The quantitative improvement is modest, but it corresponds to an adjustment in visual quality that is apparent at close range, as shown in Figure 2. Figure 3 visualizes the progress of the optimization.

Figures 1, 2, and 4 demonstrate a variety of models reconstructed with our approach. After the camera poses and the non-rigid correction functions are optimized, we compute a final color assignment for each vertex $\mathbf{p}$ using a weighted average of the corresponding colors in images $I_{\mathbf{p}}$. The corresponding color in image I_i is given weight $\mu \cos(\theta)/d^2$, where θ is the angle between the normal and the view vector at $\mathbf{p}$ for the camera that corresponds to I_i, d is the distance from $\mathbf{p}$ to the camera, and μ is a smooth matting function that assigns lower weight to image pixels that are close to image boundaries and depth discontinuities. Other integration approaches could be used [Lempitsky and Ivanov 2007; Callieri et al. 2008; Chuang et al. 2009].

Figure 4 also shows reference reconstructions produced by the widely used Point Cloud Library (PCL) [Rusu and Cousins 2011]. The geometric models and the set of key frames used for color map-

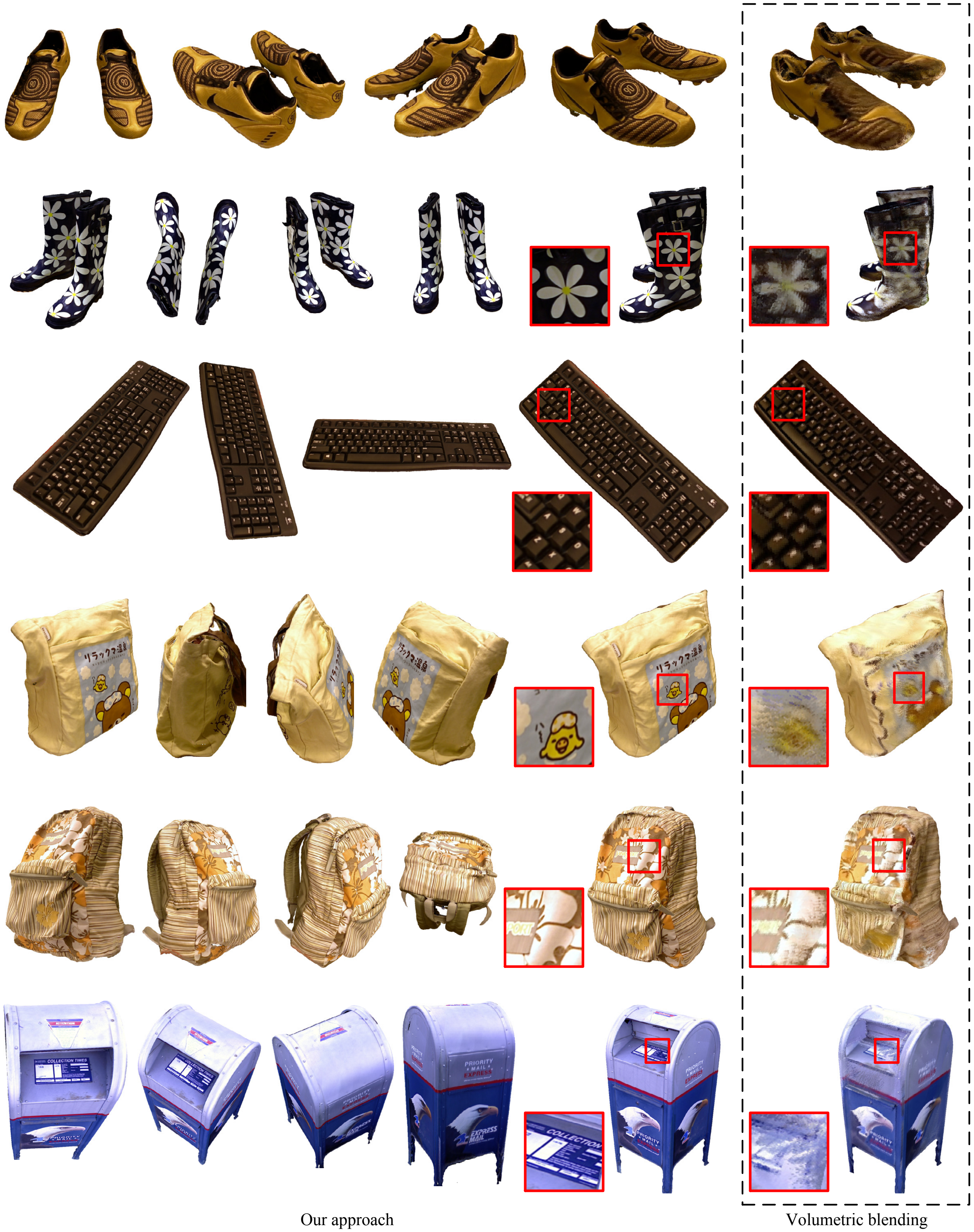


Figure 4: *A variety of additional objects reconstructed by the presented approach. For reference, the color maps produced for the same input by the widely used volumetric blending approach as implemented in the state-of-the-art Point Cloud Library are shown on the right.*

ping are the same as in the results produced by our approach. The PCL color maps are computed using volumetric blending, the approach most commonly employed in recent RGB-D reconstruction systems [Izadi et al. 2011; Nießner et al. 2013; Whelan et al. 2013; Bylow et al. 2013; Sturm et al. 2013; Endres et al. 2014]. Despite tuning the settings for maximal accuracy (512^3 voxel grid in a 0.8m^3 volume) and only using key frame images that minimize blurriness, volumetric blending fails to resolve misalignments in the data. The presented optimization approach substantially increases the visual quality of the color maps produced for these models.

The presented optimization approach successfully handles non-Lambertian objects. The fountain shown in Figure 1 is strongly non-Lambertian: specular reflection from the ceramic tiles can be clearly seen in the supplementary video. The laminated product boxes in the top row of Figure 2 are highly specular, as are the rain boots in the second row of Figure 4 and the metallic mailbox in the bottom row. To further test the sensitivity of the optimization to non-Lambertian reflectance in the scene, we scanned a highly specular ceramic pitcher from the closest range allowed by the depth camera. The result is shown in Figure 5. The optimization is resilient to non-Lambertian reflectance because it is strongly regularized: the camera pose is represented by only 6 parameters, and they are affected by the entire corresponding image. Even for highly specular objects such as the ceramic pitcher, specular highlights occupy only a small part of the image. Most of the image is reliable, and this dominates the optimization.

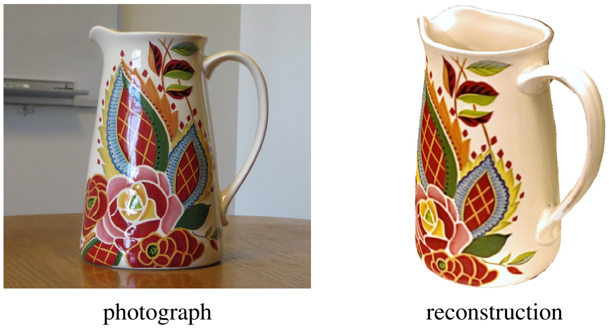


Figure 5: *A highly specular ceramic pitcher.*

To evaluate the sensitivity of the approach to error in the input camera trajectory and geometric reconstruction, we applied controlled perturbation to the camera trajectory produced by KinectFusion for the fountain model. Specifically, for each camera pose, we produce an incremental transformation represented by a 6-vector $\xi_i = (\alpha_i, \beta_i, \gamma_i, a_i, b_i, c_i)^\top$ and apply it as in equation (10). The translational and rotational components of ξ_i are sampled from zero-mean Gaussian distributions with standard deviations $\sigma_{\mathtt{t}}$ and σ_ω, respectively. We then integrate a new geometric model based on the perturbed camera trajectory, and use this model and trajectory as input to our technique. Figure 6 shows the results for $(\sigma_{\mathtt{t}} = 0.005, \sigma_\omega = 0.005)$ and $(\sigma_{\mathtt{t}} = 0.015, \sigma_\omega = 0.015)$, where $\sigma_{\mathtt{t}}$ is measured in meters.

To evaluate the sensitivity of the approach to the accuracy of the input geometric model, we progressively simplified the fountain and applied the approach to these simplified models. Figure 7 shows the resulting color maps.

7 Discussion

We presented a global optimization approach to the mapping of color images produced by consumer-grade RGB-D cameras onto the geometric models reconstructed from the corresponding range data. Our approach optimizes the camera poses for all color images along with non-rigid correction functions that help resolve misalignments that arise due to inaccurate geometry, camera localization, and optical distortions. Experimental results demonstrate that the presented approach improves the quality of reconstructed color maps.

The presented work was focused on optimizing the mapping of the color images to the reconstructed geometry. Once a consistent mapping of the images to the geometric model is computed, the images can be integrated on the model to produce a color map. The integration algorithm used in our implementation, described in Section 6, is based on weighted averaging. This is highly simplistic and leads to visual artifacts when view-dependent effects such as specular highlights or moving shadows are present during scanning. A more sophisticated integration algorithm can ameliorate such artifacts [Lempitsky and Ivanov 2007; Callieri et al. 2008; Chuang et al. 2009].

Our approach does not reason about intrinsic reflectance properties and does not produce material representations that facilitate accurate simulation of the object's appearance under different illumination conditions. The introduction of active illumination into the scene during scanning can significantly enhance the capabilities of the reconstruction system and enable the reconstruction of true surface reflectance and the creation of object models that can be accurately relit [Levoy et al. 2000; Bernardini et al. 2001; Weyrich et al. 2009]. It would be interesting to use our optimization approach in conjunction with active lighting to create high-fidelity relightable object models using consumer-grade RGB-D data. An intriguing alternative is to estimate surface reflectance without the use of active illumination [Troccoli and Allen 2008; Laffont et al. 2013; Shan et al. 2013; Chen and Koltun 2013]. Our approach can assist the application of such techniques as well.

Acknowledgements

We thank the anonymous reviewers and Philipp Krähenbühl for helpful comments and discussions.

References

Aganj, E., Monasse, P., and Keriven, R. 2009. Multi-view texturing of imprecise mesh. In *ACCV*.

Baumberg, A. 2002. Blending images for texturing 3D models. In *BMVC*.

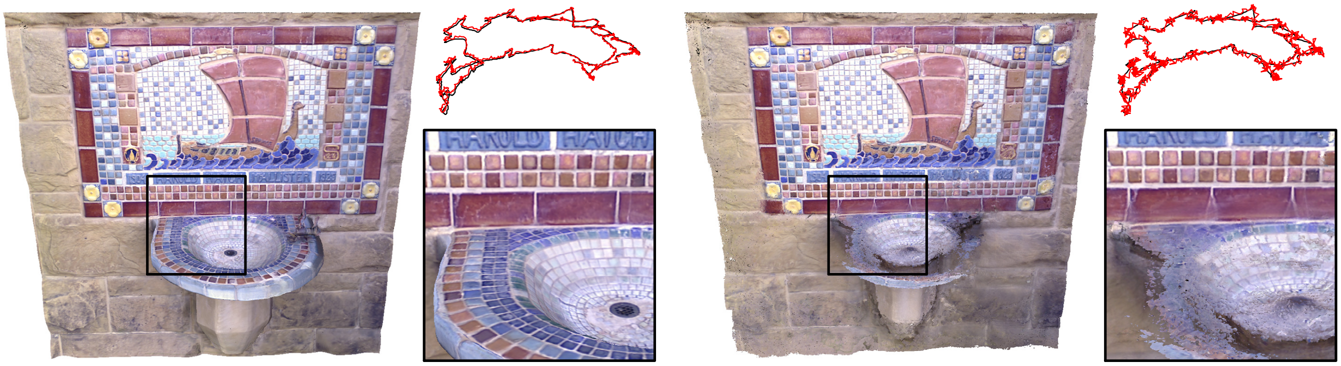
Bernardini, F., Martin, I. M., and Rushmeier, H. E. 2001. High-quality texture reconstruction from multiple scans. *IEEE Transactions on Visualization and Computer Graphics 7*, 4.

Bylow, E., Sturm, J., Kerl, C., Kahl, F., and Cremers, D. 2013. Real-time camera tracking and 3D reconstruction using signed distance functions. In *RSS*.

Callieri, M., Cignoni, P., Corsini, M., and Scopigno, R. 2008. Masked photo blending: Mapping dense photographic data set on high-resolution sampled 3D models. *Computers & Graphics 32*, 3.

Chen, Q., and Koltun, V. 2013. A simple model for intrinsic image decomposition with depth cues. In *ICCV*.

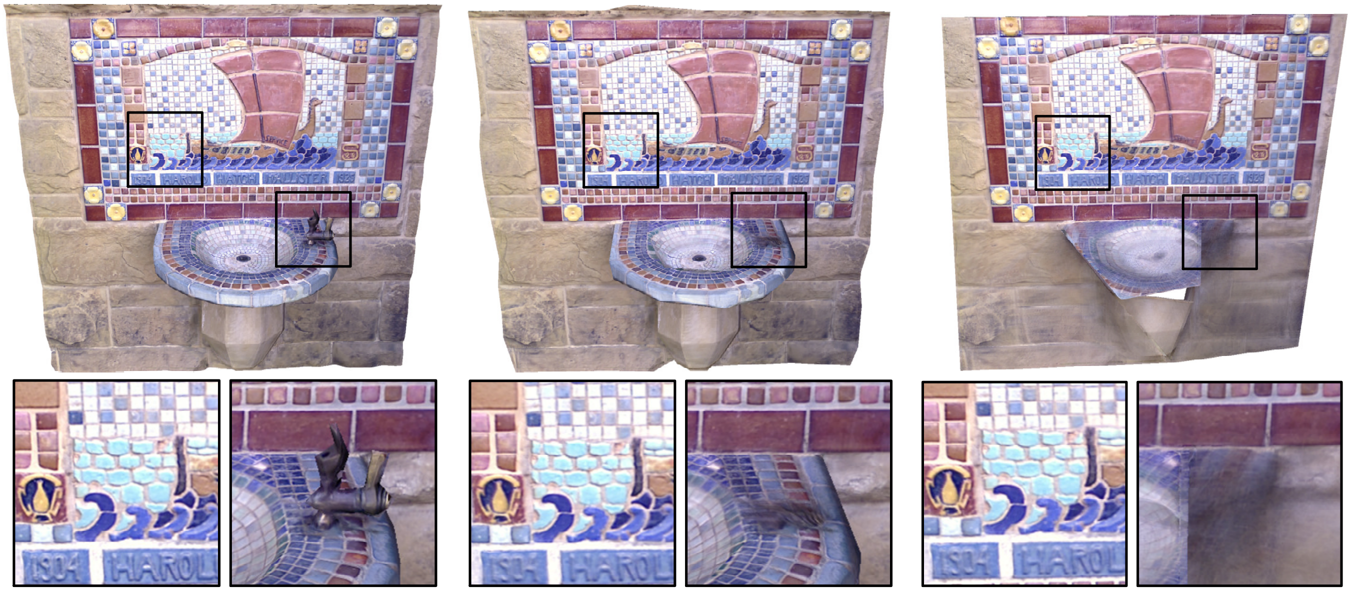
Chen, J., Bautembach, D., and Izadi, S. 2013. Scalable real-time volumetric surface reconstruction. *ACM Transactions on Graphics 32*, 4.



$\sigma_{\mathbf{t}} = 0.005, \sigma_{\omega} = 0.005$

$\sigma_{\mathbf{t}} = 0.015, \sigma_{\omega} = 0.015$

Figure 6: *Color map optimization given a perturbed camera trajectory and a corresponding geometric reconstruction. The black curves show the original camera trajectory produced by KinectFusion, the red curves show the perturbed camera trajectories. Milder perturbation on the left, stronger perturbation on the right. See text for details.*



2045 triangles

240 triangles

15 triangles

Figure 7: *Color map optimization given simplified geometry. The original model shown in Figure 1 has 990,858 triangles.*

CHUANG, M., LUO, L., BROWN, B. J., RUSINKIEWICZ, S., AND KAZHDAN, M. M. 2009. Estimating the Laplace-Beltrami operator by restricting 3D functions. *Computer Graphics Forum 28*, 5.

CORSINI, M., DELLEPIANE, M., PONCHIO, F., AND SCOPIGNO, R. 2009. Image-to-geometry registration: a mutual information method exploiting illumination-related geometric properties. *Computer Graphics Forum 28*, 7.

CORSINI, M., DELLEPIANE, M., GANOVELLI, F., GHERARDI, R., FUSIELLO, A., AND SCOPIGNO, R. 2013. Fully automatic registration of image sets on approximate geometry. *International Journal of Computer Vision 102*, 1-3.

CRETE, F., DOLMIERE, T., LADRET, P., AND NICOLAS, M. 2007. The blur effect: perception and estimation with a new no-reference perceptual blur metric. In *SPIE*.

DELLEPIANE, M., MARROQUIM, R., CALLIERI, M., CIGNONI, P., AND SCOPIGNO, R. 2012. Flow-based local optimization for image-to-geometry projection. *IEEE Transactions on Visualization and Computer Graphics 18*, 3.

ENDRES, F., HESS, J., STURM, J., CREMERS, D., AND BURGARD, W. 2014. 3D mapping with an RGB-D camera. *IEEE Transactions on Robotics 30*, 1.

FRANKEN, T., DELLEPIANE, M., GANOVELLI, F., CIGNONI, P., MONTANI, C., AND SCOPIGNO, R. 2005. Minimizing user intervention in registering 2D images to 3D models. *The Visual Computer 21*, 8-10.

GAL, R., WEXLER, Y., OFEK, E., HOPPE, H., AND COHEN-OR, D. 2010. Seamless montage for texturing models. *Computer Graphics Forum 29*, 2.

IKEUCHI, K., OISHI, T., TAKAMATSU, J., SAGAWA, R., NAKAZAWA, A., KURAZUME, R., NISHINO, K., KAMAKURA, M., AND OKAMOTO, Y. 2007. The great Buddha project: Digitally archiving, restoring, and analyzing cultural heritage objects. *International Journal of Computer Vision 75*, 1.

IZADI, S., KIM, D., HILLIGES, O., MOLYNEAUX, D., NEWCOMBE, R. A., KOHLI, P., SHOTTON, J., HODGES, S., FREEMAN, D., DAVISON, A. J., AND FITZGIBBON, A. W. 2011. KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera. In *UIST*.

JOHNSON, A. E., AND KANG, S. B. 1999. Registration and integration of textured 3D data. *Image and Vision Computing 17*, 2.

LAFFONT, P.-Y., BOUSSEAU, A., AND DRETTAKIS, G. 2013. Rich intrinsic image decomposition of outdoor scenes from multiple views. *IEEE Transactions on Visualization and Computer Graphics 19*, 2.

LEMPITSKY, V. S., AND IVANOV, D. V. 2007. Seamless mosaicing of image-based texture maps. In *CVPR*.

LENSCH, H. P. A., HEIDRICH, W., AND SEIDEL, H.-P. 2001. A silhouette-based algorithm for texture registration and stitching. *Graphical Models 63*, 4.

LEVOY, M., PULLI, K., CURLESS, B., RUSINKIEWICZ, S., KOLLER, D., PEREIRA, L., GINZTON, M., ANDERSON, S., DAVIS, J., GINSBERG, J., SHADE, J., AND FULK, D. 2000. The digital Michelangelo project: 3D scanning of large statues. In *SIGGRAPH*.

LI, H., VOUGA, E., GUDYM, A., LUO, L., BARRON, J. T., AND GUSEV, G. 2013. 3D self-portraits. *ACM Transactions on Graphics 32*, 6.

LIU, L., AND STAMOS, I. 2012. A systematic approach for 2D-image to 3D-range registration in urban environments. *Computer Vision and Image Understanding 116*, 1.

NEUGEBAUER, P. J., AND KLEIN, K. 1999. Texturing 3D models of real world objects from multiple unregistered photographic views. *Computer Graphics Forum 18*, 3.

NEWCOMBE, R. A., IZADI, S., HILLIGES, O., MOLYNEAUX, D., KIM, D., DAVISON, A. J., KOHLI, P., SHOTTON, J., HODGES, S., AND FITZGIBBON, A. 2011. KinectFusion: Real-time dense surface mapping and tracking. In *ISMAR*.

NIESSNER, M., ZOLLHÖFER, M., IZADI, S., AND STAMMINGER, M. 2013. Real-time 3D reconstruction at scale using voxel hashing. *ACM Transactions on Graphics 32*, 6.

OFEK, E., SHILAT, E., RAPPOPORT, A., AND WERMAN, M. 1997. Multiresolution textures from image sequences. *IEEE Computer Graphics and Applications 17*, 2.

PETERS, J., AND REIF, U. 1997. The simplest subdivision scheme for smoothing polyhedra. *ACM Transactions on Graphics 16*, 4.

PIGHIN, F. H., HECKER, J., LISCHINSKI, D., SZELISKI, R., AND SALESIN, D. 1998. Synthesizing realistic facial expressions from photographs. In *SIGGRAPH*.

PULLI, K., AND SHAPIRO, L. G. 2000. Surface reconstruction and display from range and color data. *Graphical Models 62*, 3.

PULLI, K., PIIROINEN, S., DUCHAMP, T., AND STUETZLE, W. 2005. Projective surface matching of colored 3D scans. In *3DIM*.

ROCCHINI, C., CIGNONI, P., MONTANI, C., AND SCOPIGNO, R. 1999. Multiple texture stitching and blending on 3D objects. In *Rendering Techniques*.

RUSU, R. B., AND COUSINS, S. 2011. 3D is here: Point Cloud Library (PCL). In *ICRA*.

SHAN, Q., ADAMS, R., CURLESS, B., FURUKAWA, Y., AND SEITZ, S. M. 2013. The visual Turing test for scene reconstruction. In *3DV*.

SINHA, S. N., STEEDLY, D., SZELISKI, R., AGRAWALA, M., AND POLLEFEYS, M. 2008. Interactive 3D architectural modeling from unordered photo collections. *ACM Transactions on Graphics 27*, 5.

STAMOS, I., AND ALLEN, P. K. 2000. 3-D model construction using range and image data. In *CVPR*.

STAMOS, I., AND ALLEN, P. K. 2002. Geometry and texture recovery of scenes of large scale. *Computer Vision and Image Understanding 88*, 2.

STURM, J., BYLOW, E., KAHL, F., AND CREMERS, D. 2013. CopyMe3D: Scanning and printing persons in 3D. In *GCPR*.

TROCCOLI, A., AND ALLEN, P. K. 2008. Building illumination coherent 3D models of large-scale outdoor scenes. *International Journal of Computer Vision 78*, 2-3.

WEYRICH, T., LAWRENCE, J., LENSCH, H. P. A., RUSINKIEWICZ, S., AND ZICKLER, T. 2009. Principles of appearance acquisition and representation. *Foundations and Trends in Computer Graphics and Vision 4*, 2.

WHELAN, T., JOHANNSSON, H., KAESS, M., LEONARD, J., AND MCDONALD, J. 2013. Robust real-time visual odometry for dense RGB-D mapping. In *ICRA*.

YAMAUCHI, H., LENSCH, H. P. A., HABER, J., AND SEIDEL, H.-P. 2005. Textures revisited. *The Visual Computer 21*, 4.

ZHOU, Q.-Y., AND KOLTUN, V. 2013. Dense scene reconstruction with points of interest. *ACM Transactions on Graphics 32*, 4.

ZHOU, Q.-Y., MILLER, S., AND KOLTUN, V. 2013. Elastic fragments for dense scene reconstruction. In *ICCV*.