Discrete-Continuous Optimization for Large-Scale Structure from Motion

David Crandall
Indiana University
Bloomington, IN
djcran@indiana.edu

Andrew Owens
MIT
Cambridge, MA
andrewo@mit.edu

Noah Snavely Dan Huttenlocher
Cornell University
Ithaca, NY
{snavely,dph}@cs.cornell.edu

Abstract

Recent work in structure from motion (SfM) has successfully built 3D models from large unstructured collections of images downloaded from the Internet. Most approaches use incremental algorithms that solve progressively larger bundle adjustment problems. These incremental techniques scale poorly as the number of images grows, and can drift or fall into bad local minima. We present an alternative formulation for SfM based on finding a coarse initial solution using a hybrid discrete-continuous optimization, and then improving that solution using bundle adjustment. The initial optimization step uses a discrete Markov random field (MRF) formulation, coupled with a continuous Levenberg-Marquardt refinement. The formulation naturally incorporates various sources of information about both the cameras and the points, including noisy geotags and vanishing point estimates. We test our method on several large-scale photo collections, including one with measured camera positions, and show that it can produce models that are similar to or better than those produced with incremental bundle adjustment, but more robustly and in a fraction of the time.

1. Introduction

Structure from motion (SfM) techniques have recently been used to build 3D models from unstructured and unconstrained image collections, including images downloaded from Internet photo-sharing sites such as Flickr [1, 6, 11, 25]. Most approaches to SfM from unstructured image collections operate incrementally, starting with a small seed reconstruction, then growing through repeated adding of additional cameras and scene points. While such incremental approaches have been quite successful, they have two significant drawbacks. First, these methods tend to be computationally intensive, making repeated use of bundle adjustment [29] (a non-linear optimization method that jointly refines camera parameters and scene structure) as well as outlier rejection to remove inconsistent measurements. Second, these methods do not treat all images equally, producing different results depending on the order in which photos are considered. This sometimes leads to failures due to local minima or cascades of misestimated cameras. Such methods can also suffer from drift as large scenes with weak visual connections grow over time.

In this paper we propose a new SfM method for unstructured image collections which considers all the photos at once rather than incrementally building up a solution. This method is faster than current incremental bundle adjustment (IBA) approaches and more robust to reconstruction failures. Our approach computes an initial estimate of the camera poses using all available photos, and then refines that estimate and solves for scene structure using bundle adjustment. This approach is reminiscent of earlier work in SfM (prior to recent work on unstructured collections) where a good initialization was obtained and bundle adjustment was used as a final nonlinear refinement step yielding accurate camera parameters and scene structure. Thus one can think of our approach as a means of providing a good initialization for highly unstructured image sets, one that is readily refined using bundle adjustment.

Our initialization technique uses a two-step process combining discrete and continuous optimization techniques. In the first step, discrete belief propagation (BP) is used to estimate camera parameters based on a Markov random field (MRF) formulation of constraints between pairs of cameras or between cameras and scene points. This formulation naturally incorporates additional noisy sources of constraint including geotags (camera locations) and vanishing points. The second step of our initialization process is a Levenberg-Marquardt nonlinear optimization, related to bundle adjustment, but involving additional constraints. This hybrid discrete-continuous optimization allows for an efficient search of a very large parameter space of camera poses and 3D points, while yielding a good initialization for bundle adjustment. The method is highly parallelizable, requiring a fraction of the time of IBA. By using all of the available data at once (rather than incrementally), and by allowing additional forms of constraint, we find that the approach is quite robust on large, challenging problems.

We evaluate our approach on several large datasets, find-

ing that it produces comparable reconstructions—and in the case of a particularly challenging dataset, a much better reconstruction—to those produced by the state-of-the-art IBA approach of [1], in significantly less time. We have also created a dataset of several thousand photos, including some with very accurate ground-truth positions taken at surveyed points. On this dataset our method and IBA have similar accuracy with respect to the ground truth, and thus our method not only can yield similar results to IBA, but the two achieve comparably accurate reconstructions.

2. Related work

Current techniques for large-scale SfM from unordered photo collections (e.g, [1, 11, 21, 25]) make heavy use of nonlinear optimization (bundle adjustment), which is sensitive to initialization. Thus, these methods are run iteratively, starting with a small set of photos, then repeatedly adding photos and refining 3D points and camera poses. While generally successful, incremental approaches are time-consuming for large image sets, with a worst-case running time $O(n^4)$ in the number of images.[1] Hence recent work has used clustering or graph-based techniques to reduce the number of images that must be considered in SfM [1, 3, 11, 26, 30]. These techniques make SfM more tractable, but the graph algorithms themselves can be costly, the number of remaining images can be large, and the effects on solution robustness are not well understood. Other approaches to SfM solve the full problem in a single *batch* optimization. These include factorization methods [28], which in some cases can solve SfM in closed form. However, it is difficult to apply factorization to perspective cameras with significant outliers and missing data (which are the norm in Internet photo collections).

Our work is most closely related to batch SfM methods that solve for a global set of camera poses given local estimates of geometry, such as pairwise relative camera poses. These include linear methods for solving for global camera orientations or translations [8, 14, 20], and L_∞ methods for solving for camera (and possibly point) positions given known rotations and pairwise geometry or point correspondence [9, 22]. While fast, these methods do not have built-in robustness to outliers, and it can be difficult to integrate noisy prior pose information into the optimization. In contrast, our MRF formulation can easily incorporate both robust error functions and priors.

Some very recent work has incorporated geotags and other prior information into SfM, as we do here. Sinha *et al*. [24] proposed a linear SfM method that incorporates vanishing points (but not geotags) in estimating camera orientations. They use only a small number of pairwise estimates of geometry (forming a spanning tree on an image graph) for initializing translations, while our method incorporates all available information. Prior information has also been used as a postprocess for SfM, e.g., by applying vanishing point or map constraints to straighten out a model [12, 23], using sparse geotags to georegister an existing reconstruction [10], or using geotags, terrain maps, and GIS data to register different connected components of a reconstruction [27]. In our work, we incorporate such geotag and vanishing point information into the optimization itself.

Finally, other techniques for accelerating SfM have been proposed, including methods for hierarchical reconstruction or bundle adjustment [7, 11, 15]. These methods still depend on an incremental approach for initialization, but structure the computation more efficiently. We present an alternative that avoids incremental reconstruction entirely.

3. Global estimation of cameras and points

Our approach represents a set of images as a graph modeling geometric constraints between pairs of cameras or between cameras and scene points (as binary constraints), as well as single-camera pose information such as geotags (as unary constraints). This set of binary and unary constraints can be modeled as a Markov random field (MRF) with an associated energy function on configurations of cameras and points. A key contribution of our work is to use both discrete and continuous optimization to minimize this energy function; in particular, we use belief propagation (BP) on a discretized space of camera and point parameters to find a good initialization, and non-linear least squares (NLLS) to refine the estimate. The power and generality of this combination of techniques allow us to efficiently optimize a more general class of energy functions than previous batch techniques. This class includes robust error functions, which are critical to obtaining good results in the presence of noisy binary and unary constraints.

3.1. Problem formulation

The input to our problem is a set of images $\mathcal{I} = \{I_1, \ldots, I_n\}$, relative pose estimates between some pairs of images (computed using two-frame SfM, described in Section 4), point correspondences between the images, and noisy absolute pose estimates for a subset of cameras (derived from sources like geotags). Our goal is to estimate an *absolute* pose for each camera, and a location for each scene point, consistent with the input measurements and in a geo-referenced coordinate system. We denote the absolute pose of camera I_i as a pair $(\mathbf{R}_i, \mathbf{t}_i)$, where $\mathbf{R}_i$ is a 3D rotation specifying the camera orientation and $\mathbf{t}_i$ is the position of the camera's optical center in a global coordinate frame. The 3D position of a scene point is denoted $\mathbf{X}_k$.

[1] If the system is dense, direct methods for solving the reduced camera matrix during bundle adjustment [29] take $O(n^3)$ time in the number of images added so far. If a constant number of images is added during each round of incremental SfM, the overall running time is $O(n^4)$. This can be alleviated for some problems through sparse or iterative methods. [1]

Each pairwise estimate of relative pose between two cameras I_i and I_j has the form $(\mathbf{R}_{ij}, \mathbf{t}_{ij})$, where $\mathbf{R}_{ij}$ is a relative orientation and $\mathbf{t}_{ij}$ is a translation direction (in the coordinate system of camera I_i). Given perfect pairwise pose estimates, the absolute poses $(\mathbf{R}_i, \mathbf{t}_i)$ and $(\mathbf{R}_j, \mathbf{t}_j)$ of the two cameras would satisfy

$$\mathbf{R}_{ij} = \mathbf{R}_i^\top \mathbf{R}_j \quad (1)$$

$$\lambda_{ij}\mathbf{t}_{ij} = \mathbf{R}_i^\top(\mathbf{t}_j - \mathbf{t}_i), \quad (2)$$

where λ_{ij} is an unknown scaling factor (due to the gauge ambiguity in 2-frame SfM). We can also write constraints between cameras and scene points. For a scene point $\mathbf{X}_k$ visible to camera I_i, let $\mathbf{x}_{ik}$ denote the 2D position of the point in I_i's image plane. Then we can relate the absolute pose of the camera and the 3D location of the point:

$$\mu_{ik}\mathbf{x}_{ik} = \mathbf{K}_i\mathbf{R}_i(\mathbf{X}_k - \mathbf{t}_i) \quad (3)$$

where $\mathbf{K}_i$ is the matrix of intrinsics for image I_i (assumed known, see Section 4), and μ_{ik} is an unknown scale factor (the depth of the point). Equation (3) is the basis for the standard *reprojection error* used in bundle adjustment. The above three constraints can be defined on a *reconstruction graph* $G = (V, E_C \cup E_P)$ having a node for each camera and each point, a set E_C of edges between pairs of cameras with estimated relative pose, and a set E_P of edges linking each camera to its visible points. Bundle adjustment typically only uses point-camera constraints, but in batch techniques constraints between cameras have proven useful.

These constraints are unlikely to be satisfied exactly because of noise and outliers in relative pose estimates, so we pose the problem as an optimization which seeks absolute poses most consistent with the constraints according to a cost function. Ideally, one would minimize an objective on camera poses and points simultaneously, as in bundle adjustment, but in practice many batch techniques solve for camera rotations and translations separately [14, 22, 24]. We follow this custom and define an MRF for each of these two subproblems. A key concern will be to use objectives that are robust to incorrect two-frame geometry and point correspondence.

Rotations. From equation (1) we see that for neighboring images I_i and I_j in the reconstruction graph, we seek absolute camera poses $\mathbf{R}_i$ and $\mathbf{R}_j$ such that $d^{\mathbf{R}}(\mathbf{R}_{ij}, \mathbf{R}_i^\top\mathbf{R}_j)$ is small, for some choice of distance function $d^{\mathbf{R}}$. This choice of distance function is tightly linked with the choice of parameterization of 3D rotations. Previous linear approaches to this problem have used a squared L_2 distance between 3×3 rotations matrices (i.e., the Frobenius norm) or between quaternions. Such methods relax the orthonormality constraints on these representations, which allows for an approximate least squares solution. In our case, we instead define $d^{\mathbf{R}}$ to be a robustified distance,

$$d^{\mathbf{R}}(\mathbf{R}_a, \mathbf{R}_b) = \rho_R(||\mathbf{R}_a - \mathbf{R}_b||), \quad (4)$$

for some parameterization of rotations (detailed below), and a robust error function ρ_R (we use a truncated quadratic).

For some cameras we may have noisy pose information from evidence like vanishing point detection and camera orientation sensors. To incorporate this evidence into our optimization, we assume that for each camera I_i there is a distance function $d_i^{\mathbf{O}}(\mathbf{R})$ that gives a cost for assigning the camera to any absolute orientation $\mathbf{R}$. This function can have any form, including uniform if no prior information is available; we propose a particular cost function in Section 4.

We combine the unary and binary distances into a total rotational error function $D^{\mathbf{R}}$,

$$D^{\mathbf{R}}(\mathcal{R}) = \sum_{e_{ij} \in E_C} d^{\mathbf{R}}\left(\mathbf{R}_{ij}, \mathbf{R}_i^\top\mathbf{R}_j\right) + \alpha_1 \sum_{I_i \in \mathcal{I}} d_i^{\mathbf{O}}(\mathbf{R}_i), \quad (5)$$

where $\mathcal{R}$ is an assignment of absolute rotations to the entire image collection, E_C is the set of camera-camera edges, and α_1 is a constant. We minimize $D^{\mathbf{R}}$ using a combination of BP and NLLS, as described in Section 4.

Camera and point positions. Having solved for camera rotations, we fix them and estimate the positions of cameras and a subset of scene points by solving another MRF inference problem on the graph G. As with the rotations, we define an error function using a combination of binary and unary terms, where the binary terms correspond to the pairwise constraints in equations (2) and (3), and the unary terms correspond to prior pose information from geotags.

Equation (2) implies that for a pair of adjacent images I_i and I_j we seek absolute camera positions $\mathbf{t}_i$ and $\mathbf{t}_j$ such that the relative displacement induced by those absolute camera positions, $\mathbf{t}_j - \mathbf{t}_i$, is close to the relative translation estimate $\hat{\mathbf{t}}_{ij} = \mathbf{R}_i\mathbf{t}_{ij}$. Similarly, for a point $\mathbf{X}_k$ visible in image I_i, we want the displacement $\mathbf{X}_k - \mathbf{t}_i$ to be close to the "ray direction" $\hat{\mathbf{x}}_{ik}$ derived from the 2D position of that point in the image (where $\hat{\mathbf{x}}_{ik} = \mathbf{R}_i^\top\mathbf{K}_i^{-1}\mathbf{x}_{ik}$ given observed position $\mathbf{x}_{ik}$ and known intrinsics $\mathbf{K}_i$). Thus, we can utilize both *camera-camera* constraints and *camera-point* constraints.

Previous linear approaches have considered one or the other of these constraints, by observing that $\hat{\mathbf{t}}_{ij} \times (\mathbf{t}_j - \mathbf{t}_i) = 0$ for camera-camera constraints [8], or that $\hat{\mathbf{x}}_{ik} \times (\mathbf{X}_k - \mathbf{t}_i) = 0$ for camera-point constraints [20]. These constraints form a homogeneous linear system, but the corresponding least squares problem minimizes a non-robust cost function that disproportionately weights distant points. Alternatively, L_∞ formulations to this problem have been defined [9, 22], but these too lack robustness. In contrast, we explicitly handle outliers by defining a robust distance on the angle between displacement vectors,

$$d^{\mathbf{T}}(\mathbf{t}_a, \mathbf{t}_b, \mathbf{t}_{ab}) = \rho(\text{angleof}(\mathbf{t}_b - \mathbf{t}_a, \mathbf{t}_{ab})), \quad (6)$$

where ρ again denotes a robust distance function.

We also integrate geotags into the optimization. For now, we simply assume that there is a cost function $d_i^{\mathbf{G}}(\mathbf{t}_i)$

for each camera I_i over the space of absolute translations, which may be uniform if a geotag is not available; we propose a particular form for $d_i^{\mathbf{G}}$ in Section 4. We define the translational error of an assignment of absolute positions $\mathcal{T}$ to the cameras as a combination of binary and unary terms,

$$D^{\mathbf{T}}(\mathcal{T}) = \alpha_2 \sum_{e_{ij} \in E_C} d^{\mathbf{T}}(\mathbf{t}_i, \mathbf{t}_j, \hat{\mathbf{t}}_{ij}) + d^{\mathbf{T}}(\mathbf{t}_j, \mathbf{t}_i, \hat{\mathbf{t}}_{ji}) \; + \\ \alpha_3 \sum_{e_{ik} \in E_P} d^{\mathbf{T}}(\mathbf{X}_k, \mathbf{t}_i, \hat{\mathbf{x}}_{ik}) + \sum_{I_i \in \mathcal{I}} d_i^{\mathbf{G}}(\mathbf{t}_i) \qquad (7)$$

where E_C denotes the set of camera-camera edges in G, E_P is the set of camera-point edges, and α_2 and α_3 are weighting constants. We could ignore one of these sets by fixing α_2 or α_3 to 0; we evaluate these options in Section 5.

3.2. Initial poses and points via discrete BP

The objectives in equations (5) and (7) can be minimized directly using Levenberg-Marquardt with reweighting for robustness, as we discuss in section 3.3, but this algorithm requires a good initial estimate of the solution. We tried using raw geotags to initialize the camera positions, for example, but we have found that they alone are too noisy for this purpose. In this section, we show how to compute a coarse initial estimate of camera poses and point positions using discrete belief propagation on an MRF.

The reconstruction graph G can be viewed as a first-order MRF with hidden variables corresponding to absolute camera orientations and camera and point positions, observable variables corresponding to prior camera pose information, and constraints between pairs of cameras and between cameras and points. Finding an optimal labeling of an MRF is NP-hard in general, but approximate methods work well on problems like stereo [4]. However compared with those problems, our MRF is highly non-uniform (dense in some places, sparse in others) and the label space is very large. To do inference on this MRF efficiently, we use discrete belief propagation (BP) [19], computing the messages in linear time using distance transforms [5]. We use BP to solve both the rotations in equation (5) and the translations in (7).

Estimating rotations. We first solve for absolute camera rotations $\mathcal{R}$ by minimizing equation (5) using discrete BP. Instead of solving for full 3D rotations, we reduce the state space by assuming that most cameras have little twist (in-plane rotation) because most photos are close to landscape or portrait orientations and most digital cameras automatically orient images correctly. (We estimate that about 80% of photos in our datasets have less than 5° twist, and 99% have less than 10° twist. The no-twist assumption is made only during the BP stage; in the later NLLS and bundle adjustment stages we allow twist angles to vary.) Under this assumption, camera orientations $\mathbf{R}_i$ can be represented as a single unit 3-vector $\mathbf{v}_i$ (the viewing direction). The distance function in equation (5) then simplifies to

$$d^{\mathbf{R}_0}(\mathbf{v}_i, \mathbf{v}_j) = \rho_R(||\mathbf{v}_{ij} - \mathbf{R}_0(\mathbf{v}_i)^{-1}\mathbf{v}_j||), \qquad (8)$$

where $\mathbf{v}_{ij}$ is the expected difference in viewing directions (which can be computed from $\mathbf{R}_{ij}$) and $\mathbf{R}_0(\mathbf{v})$ is a 3D orientation with viewing direction $\mathbf{v}$ and no twist.[2] We define $\rho_R(x) = \min(x^2, K_R)$, for constant K_R (we use 1.0).

Estimating translations and points. Having solved for absolute camera orientations, estimating camera and point positions involves minimizing Eq. (7). We use a modified pairwise distance function $d^{\mathbf{T}}$ based on the cross product between vectors, which allows us to efficiently compute BP messages using distance transforms [5]:

$$\begin{aligned} d^{\mathbf{T}}_{approx}(\mathbf{t}_a, \mathbf{t}_b, \mathbf{t}_{ab}) &= \rho_T(||\mathbf{t}_{ab} \times (\mathbf{t}_b - \mathbf{t}_a)||) \qquad (9) \\ &= \rho_T(||\mathbf{t}_b - \mathbf{t}_a||\;||\mathbf{t}_{ab}|| \sin(\theta_{ab})), \end{aligned}$$

with $\theta_{ab} = \text{angleof}(\mathbf{t}_b - \mathbf{t}_a, \mathbf{t}_{ab})$ and $\rho_T(x) = \min(x, K_T)^2$ with K_T set to about 10m. This approximation is related to the linear approach of [8], which uses a non-robust version of $d^{\mathbf{T}}_{approx}$ and estimates translations by solving a sequence of reweighted least squares problems. We note that such approaches are sensitive to outliers, as without the truncation each term is unbounded and grows with $||t_j - t_i||^2$.

3.3. Refining poses using non-linear least squares

Using the coarse estimates of rotations or translations determined by BP, we apply continuous optimization to the objective functions in equations (5) and (7), using the Levenberg-Marquardt (LM) algorithm for non-linear least squares [18]. Instead of defining a robust objective for LM, we simply remove edges and geotags from the reconstruction graph that disagree with the BP estimates more than a threshold, then run LM with a sum-of-square residual objective. These NLLS steps are related to bundle adjustment in that both minimize a non-linear objective by joint estimation of camera and (in the case of translations) point parameters. However, our NLLS stages separate rotation estimation from translation estimation, and integrate camera-camera constraints in addition to point-camera constraints.

4. A large-scale reconstruction system

We now show how to use the approach described in the last section to perform SfM on large unstructured image collections. Our method consists of the following main steps:

1. Build the reconstruction graph G through image matching and two-view relative pose estimation.
2. Compute priors from geotags and vanishing points.
3. Solve for camera orientations, $\mathcal{R}$, using discrete BP followed by continuous optimization.

[2] $\mathbf{R}_0(\mathbf{v})$ is unique unless $\mathbf{v}$ is straight up or down; such cases were uncommon enough not to have a significant effect on the optimization.

4. Estimate the camera and 3D point positions, $\mathcal{T}$, again using BP followed by continuous optimization.
5. Perform a single stage of bundle adjustment, with the pose estimates from steps 3 and 4 as initialization.

We now describe these five steps in detail.

Step 1: Producing pairwise transformations. We use SIFT matching [13] and two-frame SfM [17] to estimate correspondence and pairwise constraints between images. We tried two approaches to avoid matching all pairs of images: first, a simplification of [1] that uses a vocabulary tree [16] to find, for each image, a set of 80 candidate images to match; second, using geotags to find, for each image, 20 nearby images as initial candidates [6], sampling additional pairs from different connected components of the match graph, and densifying the graph using query expansion [1]. For matched pairs, we use the 5-point algorithm [17] followed by bundle adjustment to estimate relative pose. Since the 5-point algorithm requires intrinsically calibrated cameras, we only use images having focal lengths in the Exif metadata. We also apply a heuristic to remove high-twist images by finding images for which the relative twist of most pairwise transformations is above 20°. In addition, we remove images with unusual aspect ratios, as these are often panoramas or cropped images.

Step 2: Computing prior evidence. We compute unary cost functions on camera pose using geotags and vanishing points. For an image I_i with geotag g_i, we define the positional cost function $d_i^{\mathbf{G}}$ as a robust distance from the geotag,

$$d_i^{\mathbf{G}}(\mathbf{t}_i) = \rho_T(||\operatorname{en}(\mathbf{g}_i) - \pi(\mathbf{t}_i)||), \tag{10}$$

where ρ_T is a truncated quadratic, π is a projection of 3D camera positions into a local Cartesian plane tangent to the surface of the earth, and en maps geotags in latitude-longitude coordinates to this plane.[3] The robustified distance function is essential because geotags are typically quite noisy and contaminated with outliers [27]. For images without geotags we use a uniform function for $d_i^{\mathbf{G}}$.

For rotations, we use a cost function $d_i^{\mathbf{O}}$ for image I_i that is a sum of distances over the three rotation dimensions,

$$d_i^{\mathbf{O}}(\mathbf{R}_i) = d_i^{\theta}(\mathbf{R}_i) + d_i^{\psi}(\mathbf{R}_i) + d_i^{\phi}(\mathbf{R}_i), \tag{11}$$

where d_i^{θ}, d_i^{ψ}, and d_i^{ϕ} measure the error between an absolute camera rotation $\mathbf{R}_i$ and prior pose information in pan, twist, and tilt, respectively. For $d_i^{\phi}(\mathbf{R}_i)$, we estimate the tilt ϕ_i using vertical vanishing point (VP) detection and penalize the tilt of $\mathbf{R}_i$ as a function of angular distance to ϕ_i. We detect vertical VPs as in [24], except that we use Hough voting instead of RANSAC to find VPs. Given a vertical VP estimate with sufficient support, we compute the corresponding tilt angle ϕ_i; if no vertical VP is found, we use a uniform function for d_i^{ϕ}. To estimate pan angle we observe that equation (2) constrains the absolute orientation $\mathbf{R}_i$ of camera I_i, given absolute positions of cameras I_i and I_j and the relative translation between them. Using geotags as estimates of the camera positions, we obtain a weak cost distribution for camera pan (heading direction),

$$d_i^{\theta}(\mathbf{R}_i) = \sum_{j \in N(i)} w_i^g w_j^g \min(||\mathbf{R}_i \mathbf{t}_{ij} - \frac{g_{ij}}{||g_{ij}||}||, K_G)^2,$$

where $N(i)$ are the neighboring cameras of I_i, $g_{ij} = \operatorname{en}(\mathbf{g}_j) - \operatorname{en}(\mathbf{g}_i)$, w_i^g and w_j^g indicate whether I_i and I_j have geotags, and K_G is a constant set empirically to 0.7. Our current BP implementation assumes that cameras have zero twist (see sec. 3), so we ignore the twist error term d_i^{ψ}.

Step 3: Solving for absolute rotations. We use discrete loopy belief propagation (BP) [19] to perform inference on our MRFs. For rotations, we parameterize the unit sphere into a 3D grid with 10 cells in each dimension, for a total of $L = 1000$ labels for each camera. The advantage of this parameterization is that the distance function in equation (8) becomes separable into a sum over dimensions, which allows the use of distance transforms to compute each message in linear time [5]. (Note that cells not intersecting the surface of the unit sphere are invalid and thus are assigned infinite cost.) We then run non-linear least squares to optimize equation (4) (using a squared distance), initializing the twist angles to 0 and the viewing directions to those estimated by BP. Inside this optimization, we represent displacement rotations using Rodrigues parameters, allowing the twist angles to vary. We used Matlab's `lsqnonlin`, using its sparse preconditioned conjugate gradients solver.

Step 4: Solving for translations and points. Having estimated rotations, we next apply discrete BP to estimate camera and point positions. To reduce the label space, during BP we solve for 2D positions, as for most scenes camera and point positions vary predominantly over the two dimensions in the ground plane. (The later NLLS and BA stages remove this constraint.) We discretize this space depending on the geographic size of the region being reconstructed, using a 300×300 grid where each cell represents an area of about 1-4 meters square, for a total of $L = 90000$ labels. We use discrete BP to minimize (7) using the approximate distance function (9), with a modification to allow the use of the distance transform: when sending a message from camera i to j, instead of using the pairwise distance function $\alpha_2(d^{\mathbf{T}}(\mathbf{t}_i, \mathbf{t}_j, \hat{\mathbf{t}}_{ij}) + d^{\mathbf{T}}(\mathbf{t}_j, \mathbf{t}_i, \hat{\mathbf{t}}_{ji}))$ suggested by Eq. (7), we use $2\alpha_2 d^{\mathbf{T}}(\mathbf{t}_i, \mathbf{t}_j, \hat{\mathbf{t}}_{ij})$. For the NLLS optimization, we used `lsqnonlin` to minimize the squared residuals in Eq. (7), allowing cameras and points to vary in height as well as ground position. We generate a set of scene points by finding point tracks [1]; to reduce the size of the optimization problem, we greedily select a subset of tracks that covers each camera-camera edge in the reconstruction

[3] This coordinate frame is often called local east-north-up; we use only the 2D east and north coordinates because geotags do not include altitudes.

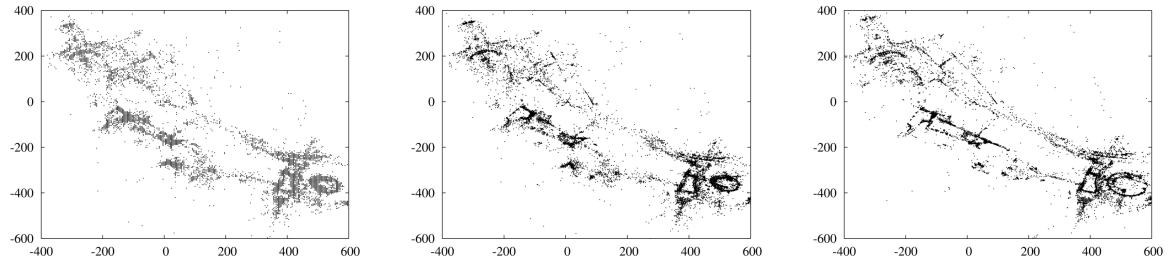

Figure 1. *Translation estimates for CentralRome.* Camera positions after BP, after NLLS refinement, and after final bundle adjustment.

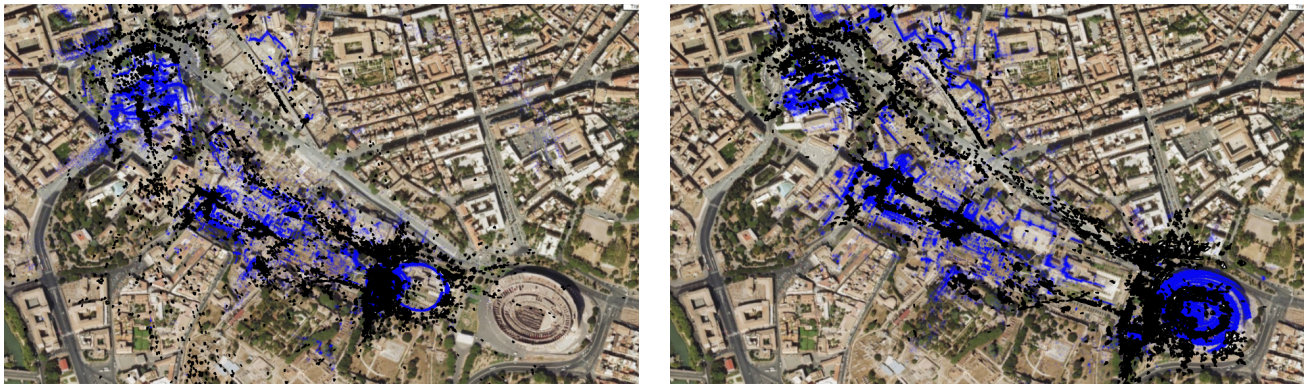

Figure 2. *CentralRome reconstruction,* using incremental bundle adjustment (left) and our technique (right), shown as top views projected on a map. Black points are cameras; blue points are scene points. There is a large drift in scale in the IBA solution (left), due to several weak connections between different parts of the reconstruction. For instance, the Colosseum (lower right) is smaller than it should be given the scale of the reconstructed Il Vittoriano monument (upper left). In addition, the inside and outside of the Colosseum do not align. The scale and alignment of the scene in our solution (right) is much more consistent.

graph at least k_1 times, and that covers each image at least $k_2 \geq k_1$ times (we used $k_1 = 5$ and $k_2 = 10$).

Step 5: Bundle adjustment. We use the estimates for the cameras and a sparse set of 3D points obtained in the last step as initialization to a global bundle adjustment stage in which all parameters including camera twist and height are refined simultaneously. We bundle adjust the cameras and the subset of 3D points selected in the previous step, triangulate the remaining points with reprojection error below a threshold, and run a final bundle adjustment. We use the preconditioned conjugate gradients bundle adjuster of [2] and a robust Huber norm on the reprojection error.

5. Results

We have applied our approach to four large datasets, summarized in Table 1, including one with over 15,000 images in the largest connected component of the reconstruction graph. The Acropolis, Dubrovnik, and CentralRome datasets consist of images downloaded from Flickr via the public API, while Quad consists of photos of the Arts Quad at Cornell University taken by several photographers over several months. For each dataset we ran the approach described in Section 4, including the discrete BP, continuous NLLS, and a final bundle adjustment. For these problems, we note that simple initializations to BA or NLLS perform poorly. We tried both random initialization of parameters, as well as initializing translations using the geotags, but both resulted in reconstructions with large errors. This highlights the fact that good initialization is critical, as well as the large degree of noise in the geotags.

Comparison to Incremental BA (IBA). To compare our approach to a state-of-the-art technique that uses IBA, we ran the datasets through a version of Bundler [25] that uses an efficient bundle adjuster based on preconditioned conjugate gradients [2], then georegistered the results by using RANSAC to align the model with the geotags. Table 2 summarizes results of this comparison, including distances between corresponding camera positions and viewing directions. It is important to note that the IBA solution has errors and is thus not ground truth, but it does represent the state-of-the-art in SfM and is thus a useful comparison. These results show that the raw geotags are quite noisy, with a median translation error of over 100 meters for some datasets. The estimates from BP are significantly better, and results from the full process (including a final bundle adjustment step) agree with the IBA solution within a meter for all datasets except CentralRome. The differences for CentralRome are large because IBA produced poor results for this set, as discussed below. The median differences between point positions for the two methods are also less than 1m for all sets except CentralRome. For the camera orienta-

Dataset	Total images	Images in largest CC ($|V|$)	Cam-cam edges ($|E_C|$)	Cam-pt edges ($|E_P|$)	% geotagged	Scene size (km^2)	Reconstructed images
Acropolis	2,961	463	22,842	42,255	100.0%	0.1×0.1	454
Quad	6,514	5,520	444,064	551,670	77.2%	0.4×0.3	5,233
Dubrovnik	12,092	6,854	1,000,178	835,310	56.7%	1.0×0.5	6,532
CentralRome	74,394	15,242	864,758	1,393,658	100.0%	1.5×0.8	14,754

Table 1. *Summary of datasets:* Total number of photos; number of images, camera-camera edges, and camera-point edges in the largest connected component; fraction of images with geotags; approximate scene size; and number of reconstructed images using our approach.

	Rotational difference					Translational difference				Point difference
	Our approach			Linear approach [8]		Our approach				Our approach
Dataset	BP	NLLS	Final BA	Linear	NLLS	Geotags	BP	NLLS	Final BA	Final BA
Acropolis	14.1°	1.5°	**0.2°**	1.6°	1.6°	12.9m	8.1m	2.4m	**0.1m**	**0.2m**
Quad	4.7°	4.6°	**0.2°**	41°	41°	15.5m	16.6m	14.2m	**0.6m**	**0.5m**
Dubrovnik	9.1°	4.9°	**0.1°**	11°	6°	127.6m	25.7m	15.1m	**1.0m**	**0.9m**
CentralRome	6.2°	3.3°	**1.3°**	27°	25°	413.0m	27.3m	27.7m	**25.0m**	**24.5m**

Table 2. Median differences between our camera pose estimates and those produced by incremental bundle adjustment.

tions, the median angle between viewing directions of the IBA solution and the output of BP is between about 5°and 14°, with the continuous optimization decreasing the difference below 5°, and the final BA step further reducing it to less than 1.5°(and below 0.5°for all datasets except CentralRome). We thus see that our approach produces reconstructions that are quantitatively similar to incremental methods in cases where IBA produces reasonable results.

We also tried the batch approach of [8] on these datasets. The rotation estimates produced by this linear technique were reasonable for the densely-connected Acropolis and Dubrovnik sets, but poor for the other two sets (as shown in the table), even when we ran NLLS on the output of [8]. The translations estimates were very poor for all of the datasets, even when we modified [8] to include geotag priors. This suggests that the robustness used by our approach is important in getting good results on large, noisy datasets (as existing evaluations of linear approaches like [8] and [24] were on much simpler, more homogeneous datasets).

Running times. As shown in Table 3, our approach is significantly faster than incremental bundle adjustment on all of the datasets that we study. The improvement is particularly dramatic for the larger datasets; for CentralRome for example, our approach took about 13 hours compared to about 82 hours for IBA, or a more than 6x speed-up. One of the reasons for this speed-up is that BP (unlike IBA) is easily parallelizable. The running times reported here used a multi-threaded implementation of rotations BP on a single 16-core 3.0GHz machine and a map-reduce implementation of translations BP on a 200-core 2.6GHz Hadoop cluster. NLLS was single-threaded and run on a 3.0GHz machine. For BA and IBA we used the highly-optimized implementation of [1], which uses a parallel BLAS library to achieve some parallelism, on a single 16-core 3.0GHz machine.

The asymptotic running time of our approach also compares favorably to that of IBA. In contrast to the the worst case $O(n^4)$ running time of IBA (using dense linear algebra), where n is the number of images, our approach is $O(n^3)$: each application of belief propagation takes time $O(n^2L)$ per iteration, where L is the size of the label space, and the final bundle adjustment step takes $O(n^3)$ time in the worst case. Memory use of BP is also $O(n^2L)$, although messages can be compressed and stored on disk between iterations (as our Hadoop implementation does).

Comparison to ground truth. To evaluate our results against ground truth, we collected highly accurate geotags (with error less than 10cm) for a subset of 348 photos for the Quad, based on survey points found using differential GPS. We also collected geotags using consumer GPS (an iPhone 3G); the precise geotags are used for ground-truth, while the consumer geotags are used as priors in the optimization.

Table 4 compares the error of camera pose estimates produced by IBA to those of the various stages of our method. IBA produces slightly better estimates than our approach, but the difference is quite small (1.01m versus 1.16m). The table also studies the sensitivity of our approach to the fraction of photos having geotags. As the fraction of geotagged images decreases below about 10%, the accuracy starts to decrease. This seems to be due to less accurate global rotation estimates, indicating that weak orientation information is helpful for getting good results. We also tested using only the camera-camera edges or only the camera-point edges during the translations estimation with 40% of images geotagged (by setting α_2 or α_3 to 0 in equation (7)); using only camera-point edges increased error from 1.21m to 1.9m, while using only camera-camera edges increased error by a factor of 3 (from 1.21m to 3.93m).

Qualitative results. Figure 1 shows views of the CentralRome dataset at different stages of our approach. Because our recovered cameras (and points) are reconstructed

Dataset	Our approach						Incremental
	Rot BP	Rot NLLS	Trans BP	Trans NLLS	Bund Adj	Total	BA
Acropolis	50s	16s	7m 24s	49s	5m 36s	**0.2 hours**	**0.5 hours**
Quad	40m 57s	8m 46s	53m 51s	40m 22s	5h 18m 00s	**7.7 hours**	**62 hours**
Dubrovnik	28m 19s	8m 28s	29m 27s	7m 22s	4h 15m 57s	**5.5 hours**	**28 hours**
CentralRome	1h 8m 24s	40m 0s	2h 56m 36s	1h 7m 51s	7h 20m 00s	**13.2 hours**	**82 hours**

Table 3. Running times of our approach compared to incremental bundle adjustment.

% geotags	BP	NLLS	Final BA
80%	7.50m	7.24m	**1.16m**
40%	7.67m	7.37m	**1.21m**
16%	7.66m	7.63m	**1.22m**
8%	8.27m	8.06m	**1.53m**
4%	18.25m	16.56m	**5.01m**

Table 4. Median error in camera position with respect to ground truth for the Quad dataset, with geotags for about 40% of images. The median error of IBA was 1.01m.

in an absolute coordinate system, they can be displayed on a map. Figure 2 shows the CentralRome reconstruction for both our approach and IBA. IBA produced a poor reconstruction for this dataset, while our approach produced much more reasonable results, likely because prior information like geotags helped to avoid problems with sparsely-connected components of the reconstruction graph.

Conclusion We have presented a new approach to SfM that avoids solving sequences of larger and larger bundle adjustment problems by initializing all cameras at once using hybrid discrete-continuous optimization on an MRF. It also integrates prior pose evidence from geotags and vanishing points into the optimization. Our approach is faster than incremental SfM both in practice and asymptotically, and gives better reconstructions on some scenes, especially when the reconstruction graph is weakly connected. As future work, we would like to further characterize the performance and tradeoffs of our algorithm, including studying its scalability to even larger collections (with hundreds of thousands of images) and characterizing its robustness to various properties of the scene and dataset. We would also like to study improvements to our approach, including solving for rotations and translations in a single optimization step.

Acknowledgments. This work was supported by the National Science Foundation (grants IIS-0705774 and IIS-0964027), the Indiana University Data to Insight Center, the Lilly Endowment, Quanta Computer, MIT Lincoln Labs, and Intel Corp., and used computational resources of the Cornell Center for Advanced Computing and Indiana University (funded by NSF grant EIA-0202048 and by IBM).

References

[1] S. Agarwal, N. Snavely, I. Simon, S. Seitz, R. Szeliski. Building Rome in a Day, *ICCV*, 2009. 3001, 3002, 3005, 3007

[2] S. Agarwal, N. Snavely, S. Seitz, R. Szeliski. Bundle Adjustment in the Large, *ECCV*, 2009. 3006

[3] F. Bajramovic, J. Denzler. Global uncertainty-based selection of relative poses for multi-camera calibrations. *BMVC*, 2008. 3002

[4] Y. Boykov, O. Veksler, R. Zabih. Fast approximate energy minimization via graph cuts, *IEEE Trans. PAMI*, 2001. 3004

[5] P. Felzenszwalb, D. Huttenlocher. Efficient belief propagation for early vision. *IJCV*, 2006. 3004, 3005

[6] J-M Frahm, P. Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y-H Jen, E. Dunn, B. Clipp, S. Lazebnik, M. Pollefeys. Building Rome on a cloudless day. *ECCV*, 2010. 3001, 3005

[7] R. Gherardi, M. Farenzena, A. Fusiello. Improving the efficiency of hierarchical structure-and-motion. *CVPR*, 2010. 3002

[8] V. Govindu. Lie-algebraic averaging for globally consistent motion estimation. *CVPR*, 2004. 3002, 3003, 3004, 3007

[9] F. Kahl, R. Hartley. Multiple-view geometry under the L-infinity norm. *IEEE TPAMI*, 2007. 3002, 3003

[10] R. Kaminsky, N. Snavely, S. Seitz, R. Szeliski. Alignment of 3D Point Clouds to Overhead Images. *CVPR Workshop on Internet Vision*, 2009. 3002

[11] X. Li, C. Wu, C. Zach, S. Lazebnik, J. Frahm. Modeling and Recognition of Landmark Image Collections Using Iconic Scene Graphs. *ECCV*, 2008. 3001, 3002

[12] P. Lothe, S. Bourgeois, F. Dekeyser, E. Royer, M. Dhome. Towards geographical referencing of monocular SLAM reconstruction using 3D city models. *CVPR*, 2009. 3002

[13] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 3005

[14] D. Martinec, T. Pajdla. Robust Rotation and Translation Estimation in Multiview Reconstruction. *CVPR*, 2007. 3002, 3003

[15] K. Ni, D. Steedly, F. Dellaert. Out-of-Core Bundle Adjustment for Large-Scale 3D Reconstruction, *ICCV*, 2007. 3002

[16] D. Nistér, H. Stewénius. Scalable Recognition with a Vocabulary Tree. *CVPR*, 2006. 3005

[17] D. Nistér. An efficient solution to the five-point relative pose problem. *Trans. PAMI*, 2004. 3005

[18] J. Nocedal, S.J. Wright. *Numerical optimization.* Springer-Verlag, New York, NY, 1999. 3004

[19] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*, Morgan Kaufmann, 1988. 3004, 3005

[20] C. Rother. Linear multi-view reconstruction of points, lines, planes and cameras, using a reference plane. *ICCV*, 2003. 3002, 3003

[21] F. Schaffalitzky, A. Zisserman. Multi-view matching for unordered image sets, or How do I organize my holiday snaps? *ECCV*, 2002. 3002

[22] K. Sim, R. Hartley. Recovering Camera Motion Using L_∞ Minimization. *CVPR*, 2006. 3002, 3003

[23] S. Sinha, D. Steedly, R. Szeliski, M. Agarwala, M. Pollefeys. Interactive 3D Architectural Modeling from Unordered Photo Collections. *SIGGRAPH ASIA*, 2008. 3002

[24] S. Sinha, D. Steedly, R. Szeliski. A multi-stage linear approach to structure from motion. *ECCV 2010 Workshop on Reconstruction and Modeling of Large-Scale 3D Virtual Environments*, 2010. 3002, 3003, 3005, 3007

[25] N. Snavely, S. Seitz, R. Szeliski. Photo tourism: exploring photo collections in 3D. *SIGGRAPH*, 2006. 3001, 3002, 3006

[26] N. Snavely, S. Seitz, R. Szeliski. Skeletal graphs for efficient structure from motion. *CVPR*, 2008. 3002

[27] C. Strecha, P. Pylvanalnen, P. Fua. Dynamic and Scalable Large Scale Image Reconstruction. *CVPR*, 2010. 3002, 3005

[28] C. Tomasi, T. Kanade. Shape and motion from image streams under orthography: a factorization method. *IJCV*, 2008. 3002

[29] B. Triggs, P. McLauchlan, R. Hartley, A. Fitzgibbon. Bundle adjustment – a modern synthesis. *Vision Algorithms: Theory and Practice*, 2000. 3001, 3002

[30] J. Vergés-Llahí, D. Moldovan, T. Wada. A new reliability measure for essential matrices suitable in multiple view calibration. *VISAPP*, 2008. 3002