TGV-SLAM: A Bayesian Approach to Real-Time Dense Monocular SLAM

Anonymous ECCV submission

Paper ID 274

Abstract. Vision based Simultaneous Localization and Mapping (SLAM) algorithms track camera motion while simultaneously estimating scene geometry from monocular RGB video. Dense SLAM algorithms have recently gained attention as an alternative to complex feature-based SLAM algorithms because they are simpler to implement, produce detailed depth maps rather than sparse point clouds, and offer constant runtime complexity. However, these algorithms often require sparse bootstrapping for initialization, or do not have a well understood theory behind initial convergence. In this paper, we derive a pure Bayesian objective for the combined problem, and show how it can be efficiently optimized, leading to a real-time algorithm for fully dense monocular SLAM that does not require sparse bootstrapping. In addition, we provide a coherent and simple theory of initial convergence, and show that the convergence of prior algorithms are explained by these same principles.

Keywords: Simultaneous Localization and Mapping (SLAM), Visual Odometry (VO), Pose Graph Optimization (PGO), Total Generalized Variation (TGV)

1 Introduction

Dense methods for visual odometry have recently gained attention as an attractive alternative to conventional feature-based simultaneous localization and mapping (SLAM) algorithms. The fundamental premise of these dense algorithms is to directly optimize for the relative camera pose parameters that register the current frame to a prior frame (given a known depth map) by minimizing photometric image differences.

Most dense methods for visual odometry utilize RGB-D cameras to obtain the depth map directly from a depth sensor [4, 5, 14, 15, 16, 17, 19, 22, 31], although dense monocular SLAM algorithms such as DTAM [21, 23] and others [27, 28], estimate a depth map by solving a regularized optimization problem utilizing poses first estimated by a sparse feature-based SFM method (e.g., PTAM [18], or directly from the five-point method [25]).

There are many disadvantage of this sparse bootstrapping approach, such as increased system complexity and overall fragility. In particular, because mapping quality is inherently sensitive to pose quality, these systems can be sensitive to the initial poses, which are estimated from limited information, and considering that future tracking quality is also sensitive to quality of the depth map, this leads to overall systems that are not self-correcting.

In contrast, a dense system that could be directly initialized on RGB video and begin simultaneously estimating a depth map while tracking relative pose would not only be algorithmically simpler (as it would eliminate the need for feature extraction, matching, estimation of constructs representing epipolar geometry and robust model fitting), but it would also be inherently self-correcting. Fairly recently, a semi-dense algorithm [11] of this type was proposed, and was used as the default method of initialization in their later work, LSD-SLAM [12], but a theory of how and why this initialization works has not been well understood.

In this paper, we explain the theory behind direct dense initialization, and present a statistically motivated, simple to implement, and computationally efficient algorithm demonstrating the power of this approach. We begin by showing how the simultaneous estimation of a depth map and tracking of a moving camera can be elegantly cast as the solution to a Bayesian objective (Section 2). For the posterior distribution, we use a rigorous Total Generalized Variation (TGV) prior (Section 2.3), that allows us to estimate *fully* dense depth maps with very low noise.

Our objective may be efficiently optimized using well-studied optimization techniques, resulting in a simple and elegant method for short-term direct SLAM (Section 3). This is modified for more extended operation with some minor indexing changes (Section 4). Our experiments (Section 5) on simulated aerial video as well as real analog video from an aerial vehicle (Section 5.1) demonstrate that the proposed algorithms are not only theoretically sound, but robust enough to produce accurate maps (Section 5.2) and poses (Section 5.3) under real-world limitations such as low resolution, analog noise and tearing, interlacing artifacts, heavy digital compression, lack of camera calibration and dynamic zooming.

2 Bayesian Objective Function

In this section we derive a Bayesian objective for maximizing the posterior probability of a joint solution encompassing camera pose and depth map (scene structure). We begin by deriving a maximum likelihood objective for the idealized problem (2.1), then correct the likelihood function to account for occlusions (Section 2.2) and finally incorporate shape priors to resolve the aperture problem, leading to our final maximum *a posteriori* (MAP) objective (Section 2.3). Our algorithm will directly minimize this objective.

2.1 Maximum Likelihood Objective

Consider a sequence of image frames $\{I^0, I^1, \ldots\}$ from a monocular camera, where $I^t \in \mathbb{R}^{N\times M}$. Because depth is inversely proportional to disparity, we wish to jointly estimate the *inverse* depth map $Z^0 \in \mathbb{R}^{N \times M}$ associated with frame 0, along with the relative pose (position and orientation) of frame t with respect to frame 0, denoted $\mathbf{p}^t \in \mathbb{R}^6$.

Let $\Omega \subset \mathbb{R}^2$ define the planar image space. Together, Z^0 and $\mathbf{p}^t$ provide an image-space mapping $W : \Omega \rightarrow \Omega$ from I^0 to I^t. For a calibrated pinhole camera, this mapping may be defined as

$$W(\mathbf{x}, Z^0, \mathbf{p}^t) = f\phi^{-1}(\mathbf{R}(\phi(\mathbf{x}/f) - Z^0(\mathbf{x})\mathbf{C})), \tag{1}$$

where $\mathbf{x} \in \Omega$, f is the focal length, $\{\mathbf{R}, \mathbf{C}\}$ are the relative orientation and position parameterized by $\mathbf{p}^t$, and $\phi : \mathbb{R}^K \rightarrow \mathbb{R}^{K+1}$ is the mapping from Euclidean coordinates into homogeneous coordinates. Note that this equation is well-defined even for infinite depths, where $Z^0(\mathbf{x}) = 0$.

Under the diffuse Lambertian surface reflectance model [20], brightness depends on surface normal and incident light ray only (i.e., unchanged by observer's angle of view). Thus, assuming the world is static, viewed through a transparent medium, and that the imaging sensor has normally distributed intensity errors, the distribution of intensity differences between matching pixels should also be normally distributed,

$$I^t(W(\mathbf{x}, Z^0, \mathbf{p}^t)) - I^0(\mathbf{x}) \sim \mathcal{N}(\sigma), \tag{2}$$

and hence the likelihood of $\{Z^0, \mathbf{p}^t\}$ from a single intensity difference is given by

$$P(Z^0, \mathbf{p}^t | \mathbf{x}, I^0, I^t) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-(I^t(W(\mathbf{x}, Z^0, \mathbf{p}^t)) - I^0(\mathbf{x}))/(2\sigma^2)\right). \tag{3}$$

For notational simplicity, we drop the subscripts and refer to Z^0 and $\mathbf{p}^t$ as simply Z and $\mathbf{p}$. The Maximum Likelihood (ML) estimates of $\{Z, \mathbf{p}\}$ are given by

$$\begin{aligned} \{\hat{Z}, \hat{\mathbf{p}}\}_{ML} &= \text{argmax}_{Z,\mathbf{p}}\, P(Z, \mathbf{p} | \mathbf{x}, I^0, I^t) && (4) \\ &= \text{argmin}_{Z,\mathbf{p}} \sum_{\mathbf{x} \in \Omega} (I^t(W(\mathbf{x}, Z, \mathbf{p})) - I^0(\mathbf{x}))^2. && (5) \end{aligned}$$

2.2 Occlusion Outliers

The minimization of (5) is sensitive to outliers of the photometric model assumed in (2), which may exist due to occlussions, moving objects, or other unmodeled effects. Assuming outliers have photometric errors uniformly distributed throughout the entire photometric range, the PDF of residuals δ in the resulting mixture distribution is given by

$$p_{mix}(\delta) = m\exp(-\delta^2) + (1-m)\,(1/(b-a))\,, \tag{6}$$

where $m \in (0,1)$ is the fraction of outliers and (a,b) is the photometric range (e., 0-255). Thus, the effect of outliers may be compensated for by using a robust weighting function of

$$w(\delta) = p_i^{-1}(p_{mix}(\delta)) = \sqrt{-\ln\left(m\exp(-\delta^2) + (1-m)\,(1/(b-a))\right)}/|\delta|. \tag{7}$$

A concern with (7) is that $\lim_{\delta\to 0} w(\delta) = \infty$, which could destabilize the solution by putting too much weight on a few good inliers. A function that very closely approximates (7), with the exception of removing this unstable singularity is given in [13, §A6.8] by

$$w_{BZ}(\delta) = \begin{cases} 1, & |\delta| < \tau \\ \tau/|\delta|, & \text{o/w}, \end{cases} \tag{8}$$

where τ is determined from the mixing ratio as $\tau = \sqrt{-\ln((1-m)/(b-a))}$. In practice, because the true mixing ratio is unknown, we compute τ from the standard deviation of photometric inliers. For example, if α is the desired percentile of inliers to capture (e.g., $\alpha = 0.95$), then $\tau = \sigma_i\sqrt{\Phi^{-1}(\alpha)}$, where Φ is the CDF of the standard normal function.

2.3 Regularization

Let $u \in \mathbb{R}^{N\times M}$ represent the inverse depth map to be estimated (previously denoted by Z). Then the objective so far is to minimize

$$F(u) = \sum_{\mathbf{x}\in\Omega} (I^t(W(\mathbf{x}, u, \mathbf{p})) - I^0(\mathbf{x}))^2 w_{BZ}(\delta). \tag{9}$$

However, in order to overcome the aperture problem, it is important to incorporate some prior PDF for the shape of u, which we denote $P(u)$. If we denote the negative log of the prior (regularization function) as $\Psi(u) = -\log P(u)$, we obtain the regularized problem

$$\min_u \lambda F(u) + \Psi(u), \tag{10}$$

where λ is a relative weight on the data term.

Without making particular assumptions about *what* is being observed by the camera, the least limiting prior is to assume some level of continuity in the inverse depth map u. As demonstrated by other recent methods [28, 29], one of the most realistic and effective known priors for depth maps is the Total Generalized Variation (TGV) [3] of order 2. Geometrically, the TGV of order 2 seeks a piecewise-planar solution, and excels at estimating planar or smoothly varying surfaces while still admitting discontinuities in the field. In its primal form, the TGV of order $k = 2$ is given by

$$TGV_{\alpha}^{2}(u) = \min_{w \in \mathbb{R}^2} \left\{ \alpha \int_{\Omega} |\nabla u - w| \, dx + \int_{\Omega} |\mathcal{E}(w)| \, dx \right\}. \tag{11}$$

Substituting (11) into (10), we obtain our joint maximum *a posteriori* (MAP) objective,

$$E(Z, \mathbf{p}) = \lambda F(u) + TGV_{\alpha}^{2}(u). \tag{12}$$

3 Optimization

We now discuss the efficient minimization of (12). This is accomplished by alternating minimization of map and pose (Section 3.1). We then discuss initialization (Section 3.2), and present algorithms for the sub-optimization of pose (Section 3.3) and inverse depth map (Section 3.4).

3.1 Adaptive Alternating Minimization

Alternating Minimization (AM) [6, 7, 10] is a very general framework for nonlinear minimization of functions of two variables that alternates between holding one variable fixed and optimizing the other. There are many well-known practical applications, such as Expectation Maximization (EM).

Considering that our objective (12) is essentially a function of two 'variables' (an inverse depth map Z, and a relative pose $\mathbf{p}$) and that the estimation of either depth map given known relative pose or of relative pose given known depth map are both well-studied problems, AM might seem to be a natural optimization choice. However, both dense photometric tracking algorithms and dense-stereo correspondence algorithms are highly sensitive to each other. As such, initialization would present somewhat of a chicken-and-egg problem that we believe explains why AM is not a standard approach to the dense SLAM problem (as it is for sparse SLAM).

Ignoring the initialization problem for now, a limitation of the above AM method is that we would only be able to estimate map and pose for a single view pair, whereas in the SLAM problem we wish to estimate a sequence of continuous camera poses. An adaptive variation of the AM algorithm was studied in Niesen et al. [24], where it was found to be capable of converging to a running estimator of the latest parameter values when the parameters to be estimated are continuous. This adaptive variation may be applied to the SLAM problem by letting one variable be a fixed reference depth map, and the other variable represent the continuously varying camera parameters (Algorithm 1). Of course, this algorithm will only be applicable so long as the current camera view retains some observability of the map – a limitation that we later relax with the Leapfrog algorithm (Section 4).

Algorithm 1 SLAM by Adaptive Alternating Minimization

```
Ensure: Estimation of $\{Z^0, \mathbf{p}^0, \mathbf{p}^1, \mathbf{p}^2 \dots\}$
  $\{Z^0, \mathbf{p}^0\} \leftarrow$ initialize
  $t \leftarrow 1$
  repeat
      $\mathbf{p}^t \leftarrow \operatorname{argmin}_{\mathbf{p}} E(Z^0, \mathbf{p})$        ▷ Estimate latest pose
      $Z^t \leftarrow \operatorname{argmin}_Z E(Z, \mathbf{p}^t)$        ▷ Update map estimate
      $t \leftarrow t + 1$        ▷ Next frame
  until no more frames
```

3.2 Initialization

Despite the aforementioned difficulty of initializing AM for a single view pair (Section 3.1), it turns out that initial convergence of *adaptive* AM SLAM (Algorithm 1) is much more stable, and we obtain good results by simply starting from a canonical relative pose and constant valued inverse depth map ($Z = 1$).

To understand why this initialization works, consider that in the limit as baseline goes to zero, motion parallax effects disappear and the image warping may be perfectly described by a planar homography. In Algorithm 1, we start tracking immediately, while motion parallax effects are negligible and the relative pose change is nearly identity. This is crucial to the algorithm, because if the images do not provide enough information to *estimate* a depth map, then they also must not provide enough information to *disagree* with an inaccurate depth map. As such, we are free to initialize the inverse depth map as a fronto-parallel plane (e.g., constant valued). Because the scale of the pose updates is only defined relative to the scale of the depth map, we use the $Z = 1$ plane without loss of generality.

Because the initial map is planar, the initial tracking step effectively performs photometric image registration using a planar homography (parameterized by camera pose parameters). Even though the scene is not truly a fronto-parallel plane, this transformation still has sufficient generality to register the images well at small baseline. Although the 3D position of this first tracked pose will not be precise in a geometric sense, the resulting epipolar lines in image space are generally quite good, typically having epipolar line orientation errors of less than a few degrees. Moreover, at low baseline, the epipolar restricted search is much less sensitive to epipolar line orientation errors, because the total disparity (eg, movement along the epipolar line) will only be a few pixels at most. Finally, because the motion parallax effects are initially negligible, the epipolar search need not be exhaustive along the epipolar line – when there is no motion parallax, correspondences will be described exactly by the plane and pose parameters. Thus, it is only necessary to search along the direction of the epipolar line within the current local photometric basin of attraction. This justifies our algorithm for local nonlinear *refinement* of depth (Section 3.4) as opposed to using a more global depth search. Indeed, we find that restricting the epipolar search to the local basin of attraction is in fact necessary for reliable convergence.

Because the baseline is initially quite small, the initial 'depth maps' will have very low precision. They will show only marginal depth relief, with approximately one discreet depth value for each whole pixel unit of disparity. The use of regularization can smooth out this estimate, effectively interpolating depth values inbetween those discrete depth planes, but it will still suffer from an inevitable loss of dynamic range that produces a squashed effect. Nonetheless, the additional depth relief gained via matching tends to precede the required level of depth relief required for tracking, so that as the baseline increases with each successfully tracked frame, so does the relief of the depth map increase in the next mapping iteration.

In our tests, we typically observed that the algorithm converges to a good joint estimate of depth map and pose within the first second of 30 fps video (Fig. 1), and from there it proceeds to surf the changing photometric minima of the pose as the camera continues to move, thereby solving the SLAM problem and the initialization problem simultaneously.

3.3 Pose Update Step

The map regularization term drops out of the pose update step, so the problem reduces to the weighted ML estimation of camera pose given a fixed map by direct minimization of photometric residual,

$$\begin{aligned} \hat{\mathbf{p}}^t_{ML} &= \operatorname{argmin}_{\mathbf{p}} E(Z^0, \mathbf{p}^t) && (13) \\ &= \operatorname{argmin}_{\mathbf{p}} \sum_{\mathbf{x} \in \Omega} (I^t(W(\mathbf{x}, Z^0, \mathbf{p}^t)) - I^0(\mathbf{x}))^2 w(\delta). && (14) \end{aligned}$$

We solve (14) using the forward-additive approach [1]. To summarize, this involves first linearizing (14) using a first-order Taylor expansion, then solving the 6-dimensional linear system for the parameter update that would take the partial derivatives of the linearized estimate to zero. Thus, if we denote the Jacobian matrix as $\mathbf{J}$, parameter update as $\Delta\mathbf{p}$, and errors as E, we must solve the following linear system:

$$\sum_{\mathbf{x}} \mathbf{J}(\mathbf{x})^\mathsf{T} \mathbf{J}(\mathbf{x}) \Delta\mathbf{p} = \sum_{\mathbf{x}} \mathbf{J}(\mathbf{x})^\mathsf{T} E(\mathbf{x}). \quad (15)$$

Iteratively solving (15) leads to a solution of (14) via the Gauss-Newton method. Standard enhancements such as Levenberg-Marquardt or Powell's dog leg [30] may also be applied, but we find a simple Gauss-Newton method with line search in the update direction to be most efficient in this case.

To improve the basin of convergence, we re-solve the problem in a coarse to fine approach using an image pyramid. In order to obtain realtime performance, we compute all the residuals and partial derivatives in CUDA, and sum up all the contributions to the final linear system using a custom reduction kernel, so that the entire computation is performed in parallel on the GPU.

3.4 Map Update Step

The term dependent on pose parameters drops out of the map update step, and the remaining problem is to estimate the maximum *a posteriori* inverse depth map given a fixed estimate of relative pose,

$$\hat{Z}^0_{MAP} = \operatorname{argmin}_Z E(Z, \mathbf{p}^t) \tag{16}$$

$$= TGV^2_\alpha(u) + \lambda F(u). \tag{17}$$

We solve (17) using the primal dual method [9], a recently developed variational method that can efficiently solve non-differentiable saddle point problems. Applying the primal dual method to the TGV problem [2, 34] leads to a nonlinear algorithm for regularized inverse depth map estimation (Algorithm 2).

Algorithm 2 PD algorithm for regularized depth map estimation

```
 1: $u \leftarrow$ initial estimate
 2: $w, p, q \leftarrow 0$
 3: repeat
 4:     $p \leftarrow \mathrm{proj}_{\alpha_1}(p + \sigma(\nabla u - w))$
 5:     $q \leftarrow \mathrm{proj}_{\alpha_0}(q + \sigma\mathcal{E}(w))$
 6:     $\bar{u} \leftarrow \mathrm{prox}^\tau_F(u - \tau\nabla^\dagger p)$
 7:     $\bar{w} \leftarrow w + \tau(p - \mathcal{E}^\dagger q)$
 8:     $u \leftarrow 2\bar{u} - u$
 9:     $w \leftarrow 2\bar{w} - w$
10: until converged
```

The first two steps of Algorithm 2 (lines 4-5) improve the dual variables p, q, the next two steps (lines 6-7) improve the primal variables u, w, and the final two steps (lines 8-9) are a forward extrapolation that helps speedup convergence. In this case, the proximal operators are defined by

$$\mathrm{proj}_{\alpha_1}(\bar{p}) = \frac{\bar{p}}{\max(1, ||\bar{p}||/\alpha_1)} \quad \text{and} \quad \mathrm{prox}^\tau_{\mathcal{F}}(\bar{u}) \qquad = \frac{\bar{u} - \tau\lambda ab}{1 + \tau\lambda a^2}, \tag{18}$$

where $a = \frac{\partial I^t}{\partial u}$ and $b = I^t(\mathbf{x}') - \bar{u}\frac{\partial I^t}{\partial u} - I^0(\mathbf{x})$. Further details of this derivation may be found in the supplementary material.

Parameter Selection A known requirement for convergence is that $\sigma\tau \leq 1/\left|\left|\mathcal{K}\right|\right|^2$, where in our case it may be verified that $\left|\left|\mathcal{K}\right|\right|^2 < 12$. It is often recommended to use $\sigma = \tau = 1/\left|\left|\mathcal{K}\right|\right|$ [9], but a key element to the success of our implementation was to relate the step sizes to physically significant scale factors.

To this end, it may be verified that τ is proportional to the primal step size in Z, so $\tau \propto Z$. Furthermore, we know that $\tau \propto 1/\sigma$, so it follows that $\sigma \propto 1/Z$. Finally, from (12) it may be verified that $\lambda \propto Z/I^2$, where I is the range of image intensity. Taking all of these proportionality constraints together implies that, in order to remain independent of the reconstructed 3D scale or image intensity scale, one should choose

$$\tau = Z/(A\sqrt{12}) \quad \text{and} \quad \sigma = A/(Z\sqrt{12}) \quad \text{and} \quad \lambda = (Z/I^2)B, \tag{19}$$

for some constants A and B. In practice, we find using TGV parameter $\alpha = 2$ with $A = 300$ and $B = 5$ result in speedy convergence for all scenes we have tested.

4 Leapfrog Algorithm

Because the alternating minimization SLAM algorithm (Algorithm 1) only estimates a single depth map corresponding to the first frame of video, it can only be used to track the camera while there is significant visual overlap with the original frame. Thus, in order to track reliably for long periods of time with less restrictions about how the camera moves, it will be necessary to periodically update the tracking reference map. We accomplish this using a leapfrog algorithm (Algorithm 3). When used with the proposed photometric pose update (Section 3.3) and TGV mapping update (Section 3.4), we refer to this as TGV-SLAM.

The basic idea is to decouple the *tracking reference* (the depth map that the relative pose is estimated with respect to) from the *mapping reference* (the depth map that is updated). Initially, the tracking and mapping reference are both set to frame 0. As new frames come in, tracking and mapping is continued until the map converges to sufficient quality, at which point the mapping reference is advanced: that is, start mapping a new (more recent) frame without changing which frame we are tracking from. Thus, as the camera moves over time, new maps are produced from various incremental positions along the track. When the tracking baseline becomes too large and tracking becomes unreliable, switch the tracking reference to the next oldest mapping reference.

Our approach differs from that of LSD-SLAM, where tracking and mapping reference were always the same, and updating the tracking reference was done by warping the last estimated depth map into the latest frame and resetting the tracking baseline to zero [12]. We observe a number of downsides to that approach: (1) it leads to periodic reductions in tracking accuracy, because after the baseline is reset there is insufficient baseline to accurately identify relative pose; (2) map estimation also becomes less constrained after a baseline reset, resulting in scale drift that needed to be explicitly handled in the Pose Graph

Algorithm 3 Leapfrog algorithm for Extended SLAM

```
$trackInd \leftarrow 0$
$mapInd \leftarrow 0$
$curInd \leftarrow 1$
$queue \leftarrow empty$
repeat
    repeat
        track relative pose of $curInd$ from $trackInd$
        if tracking failed then
            if $queue$ is empty then return                                ▷ abort
            else
                $trackInd \leftarrow$ pop the top of $queue$
            end if
        end if
    until tracking succeeds
    update map at $mapInd$ using frame $curInd$
    if map has converged then
        push $mapInd$ onto $queue$
        $mapInd \leftarrow curInd$
    end if
    $curInd \leftarrow curInd + 1$
until no more frames
```

Optimizer (PGO); (3) errors in the map may be propagated forward into future maps, where those errors may be magnified.

By contrast, Algorithm 3 always tracks from the oldest keyframe from which tracking may be successfully performed, thereby ensuring that the baseline is wide and tracking precision is not degraded (once initialization is complete). Because the mapping baseline is never reset to zero, no scale drift is introduced. As such, there is no explicit need for pose graph optimization. In addition, it minimizes drift by not creating excessive keyframes. Keyframes are only created when the camera motion dictates it. Thus, a camera that hovers around the same area might never create additional keyframes and thereby never accumulate drift.

5 Experiments

5.1 Datasets

In order to independently test algorithm components and empirically assess tracking and estimated depth map accuracy, it is useful to have test video with associated ground truth depth maps and camera poses. To this end, we first validate algorithm performance on rendered imagery that simulates an orbit with central stare over Wilson Canyon, NV. This was generated using USGS Digital Elevation Models (DEMs) [33] and textures and rendered using osgEarth [26], using an orbit radius of 1000 meters, a flight speed of about 70 m/s, and camera frame rate of 30 fps.

In addition, we validate the algorithm on aerial video acquired from an analog broadcast by a vehicle orbiting over a factory in Boardman, OR (approx. 2,300 meter radius). For evaluation purposes, truth poses were generated from coupled GPS/IMU telemetry and precisely interpolated to provide frame-level estimates of pose. Due to reasons out of our control, the camera was *not* calibrated for intrinsic parameters (such as radial distortion or principal point), although a rough estimate of the sensor focal length is provided in the telemetry, which is variable over time (due to dynamic zooming).

5.2 Map Convergence

We begin by showing the initial convergence of an inverse depth map on both the simulated Wilson Canyon dataset and also the real Boardman video, with a comparison to LSD-SLAM for reference (Fig. 1). On both datasets with TGV-SLAM, some vague depth information begins to resolve within the first 5 frames; by frame 10, a blurry depth map is evident; by frame 40, details and discontinuities emerge. By contrast, LSD-SLAM does not begin to depart from random noise until about frame 20, and the final result is much noiser, as it lacks a rigorous method for regularized map estimation. Additional videos demonstrating convergence may be found in the supplementary material.

Running for a longer duration, TGV-SLAM produces inverse depth maps at various keyframes (Fig. 2). We note that it is often difficult to discern any visual difference between the estimated depth maps and the ground truth depth maps. Although we lack ground truth for the Boardman scene, we note that visual fidelity of the depth maps appear consistent with the imagery, and even succeed in tracking through dynamic zoom changes (Fig. 3). A 3D model was produced for the Boardman scene (Fig. 4) by volumetric fusing of the keyframe depth maps using a signed distance function (similar to [8]), and then synthesizing a texture map from the input imagery.

5.3 Pose Accuracy

Relative pose accuracy is assessed by aligning the computed relative poses to the truth camera centers using the least-squares similarity transform [32]. Our results on the Wilson dataset (Fig. 5) show that the mean relative error from TGV-SLAM (0.041861) was 53% less than that of LSD-SLAM (0.07818). For comparison, we also plotted the error due to pose updates only from a single truth depth map (Fig. 5). In this case, the mean relative error was only 0.000114 (nearly 400 times less), indicating that the primary source of remaining pose error in TGV-SLAM is due to biases in the estimated depth maps (despite the apparently good visual fidelity in Fig. 2).

Relative pose accuracy (as measured from telemetry) for TGV-SLAM on the Boardman video was significantly better (Fig. 6), until it eventually destabilizes due to rapid dynamic zooming of the camera. We provide no comparison to LSD-SLAM because it was not able to converge, likely due to the large scene scale (2,300 meter distance) and narrow field of view (initially 8.8 degrees).

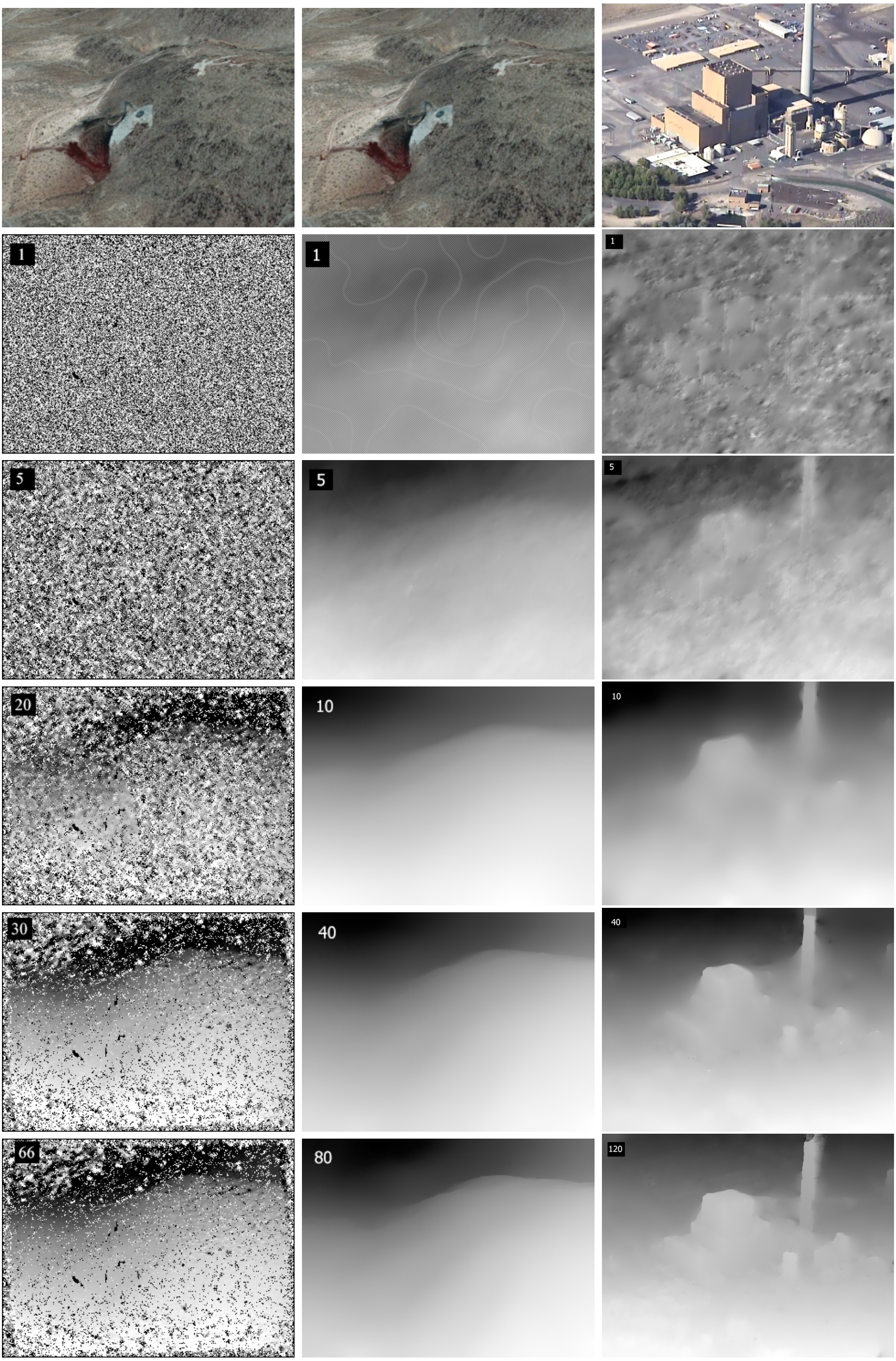


Fig. 1. Convergence of inverse depth map as a function of frame number (indicated in the upper left corners of each map) in 30 fps video. The first frame of each sequence is shown in top row. Left column: LSD-SLAM on the Wilson scene (for reference); Middle column: TGV-SLAM on the Wilson scene; Right column: TGV-SLAM on Boardman scene.

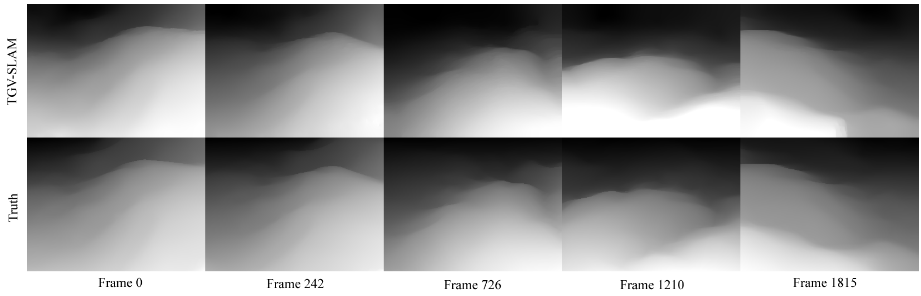


Fig. 2. Top row: TGV-SLAM keyframe inverse depth maps on Wilson scene. Bottom row: true depth maps for comparison.

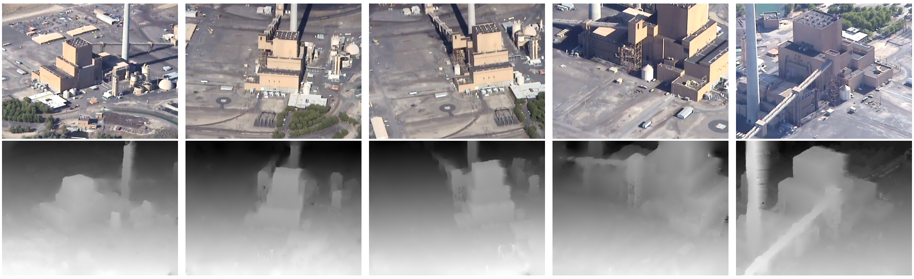
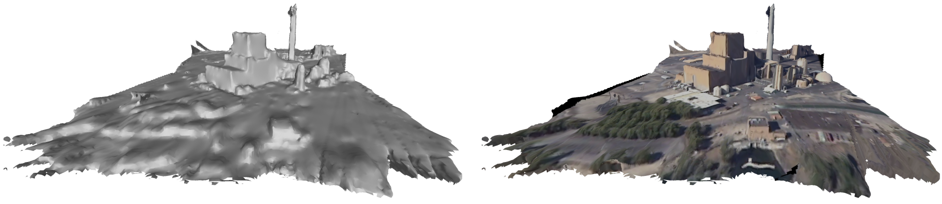


Fig. 3. Selected keyframes and associated inverse depth maps estimated by TGV-SLAM on the Boardman dataset.



(a) (b)

Fig. 4. Volumetric 3D reconstruction of Boardman scene. (a) without texture; (b) with texture synthesized from the video.

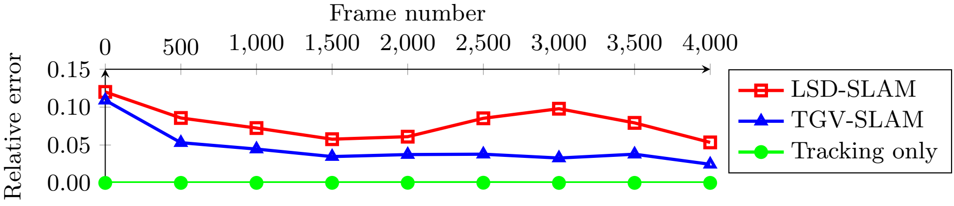


Fig. 5. Relative pose errors (from truth) on the Wilson Canyon simulated video.

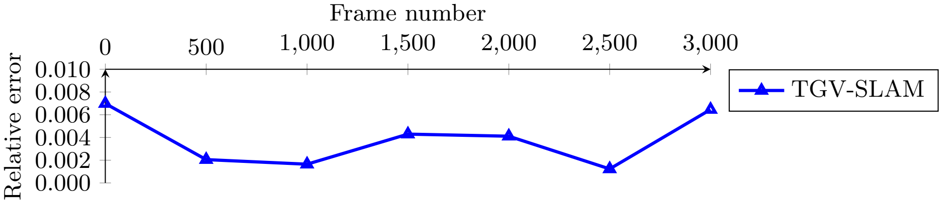


Fig. 6. Relative pose errors (from telemetry) on the Boardman video. LSD-SLAM (not shown) was unable to converge.

6 Conclusions

Dense methods for monocular SLAM present an elegant alternative to traditional feature-based methods, offering simple implementations and constant time frame processing. However, they typically resort to rough approximations and heuristics, and a theory behind direct initialization has been lacking. In addition, they typically require fixed zoom cameras, good calibration, wide field of view sensors and small scale scenes. In this report we have demonstrated that dense methods can also be effective on large scale scenes with narrow field of view sensors and dynamic zoom, even from noisy, poorly calibrated analog video streams. More importantly, we have elucidated the principles behind the initial convergence of direct dense initialization, and shown how to derive and implement a simple SLAM algorithm by directly optimizing a Bayesian objective.

References

[1] Baker, S., Matthews, I.: Lucas-kanade 20 years on: A unifying framework. International Journal of Computer Vision 56(3), 221–255 (2004)

[2] Bredies, K.: Recovering piecewise smooth multichannel images by minimization of convex functionals with total generalized variation penalty. In: Bruhn, A., Pock, T., Tai, X.C. (eds.) Efficient Algorithms for Global Optimization Methods in Computer Vision, Lecture Notes in Computer Science, vol. 8293, pp. 44–77. Springer Berlin Heidelberg (2014)

[3] Bredies, K., Kunisch, K., Pock, T.: Total generalized variation. SIAM Journal on Imaging Sciences 3(3), 492–526 (2010)

[4] Bylow, E., Sturm, J., Kerl, C., Kahl, F., Cremers, D.: Direct camera pose tracking and mapping with signed distance functions. In: Demo Track of the RGB-D Workshop on Advanced Reasoning with Depth Cameras at the Robotics: Science and Systems Conference (RSS) (June 2013)

[5] Bylow, E., Sturm, J., Kerl, C., Kahl, F., Cremers, D.: Real-time camera tracking and 3d reconstruction using signed distance functions. In: Robotics: Science and Systems Conference (RSS) (June 2013)

[6] Byrne, C.L.: Alternating minimization and alternating projection algorithms: A tutorial. Tech. rep., Department of Mathematical Sciences, University of Massachusetts Lowell, Lowell, MA (March 2011)

[7] Byrne, C.L.: Alternating minimization as sequential unconstrained minimization: A survey. Journal of Optimization Theory and Applications 156(3), 554–566 (2013)

[8] Calakli, F., Taubin, G.: Expanding the Frontiers of Visual Analytics and Visualization, chap. SSD-C: Smooth Signed Distance Colored Surface Reconstruction, pp. 323–338. Springer London, London (2012)

[9] Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. Journal of Mathematical Imaging and Vision 40(1), 120–145 (2011), `http://dx.doi.org/10.1007/s10851-010-0251-1`

[10] Csiszár, I., Tusnády, G.: Information Geometry and Alternating minimization procedures. Statistics and Decisions Supplement Issue 1 (1984)

[11] Engel, J., Sturm, J., Cremers, D.: Semi-dense visual odometry for a monocular camera. In: Computer Vision (ICCV), 2013 IEEE International Conference on. pp. 1449–1456 (Dec 2013)

[12] Engel, J., Schöps, T., Cremers, D.: Lsd-slam: Large-scale direct monocular slam. In: Computer Vision–ECCV 2014, pp. 834–849. Springer (2014)

[13] Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, New York, NY, USA, 2 edn. (2003)

[14] Kerl, C.: Odometry from RGB-D Cameras for Autonomous Quadrocopters. Master's thesis, Technical University Munich, Germany (Nov 2012)

[15] Kerl, C., Sturm, J., Cremers, D.: Dense visual slam for rgb-d cameras. In: Proc. of the Int. Conf. on Intelligent Robot Systems (IROS) (2013)

[16] Kerl, C., Sturm, J., Cremers, D.: Robust odometry estimation for rgb-d cameras. In: Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA) (May 2013)

[17] Kerl, C., Sturm, J., Cremers, D.: Robust odometry estimation for rgb-d cameras. In: ICRA (2013)

[18] Klein, G., Murray, D.: Parallel tracking and mapping for small AR workspaces. In: Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07). Nara, Japan (November 2007)

[19] Klose, S., Heise, P., Knoll, A.: Efficient compositional approaches for real-time robust direct visual odometry from rgb-d data. In: Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on. pp. 1100–1106. IEEE (2013)

[20] Lambert, J.H.: Photometria sive de mensure de gratibus luminis, colorum umbrae. Eberhard Klett (1760)

[21] Newcombe, R.A., Davison, A.J.: Live dense reconstruction with a single moving camera. In: The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010. pp. 1498–1505 (2010)

[22] Newcombe, R.A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A.J., Kohli, P., Shotton, J., Hodges, S., Fitzgibbon, A.W.: Kinectfusion: Real-time dense surface mapping and tracking. In: 10th IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2011, Basel, Switzerland, October 26-29, 2011. pp. 127–136 (2011)

[23] Newcombe, R.A., Lovegrove, S., Davison, A.J.: DTAM: dense tracking and mapping in real-time. In: IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011. pp. 2320–2327 (2011)

[24] Niesen, U., Shah, D., Wornell, G.W.: Adaptive alternating minimization algorithms. IEEE Trans. Inf. Theor. 55(3), 1423–1429 (Mar 2009)

[25] Nistér, D.: An efficient solution to the five-point relative pose problem. IEEE Trans. Pattern Anal. Mach. Intell. 26(6), 756–777 (Jun 2004), http://dx.doi.org/10.1109/TPAMI.2004.17

[26] Pelican Mapping: osgearth: Geospatial sdk for openscenegraph (2015), http://osgearth.org

[27] Piniés, P., Paz, L.M., Newman, P.: Dense and Swift Mapping with Monocular Vision. In: International Conference on Field and Service Robotics (FSR). Toronto, ON, Canada (June 2015)

[28] Piniés, P., Paz, L.M., Newman, P.: Dense Mono Reconstruction: Living with the Pain of the Plain Plane. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). Seattle, WA, USA (May 2015)

[29] Ranftl, R., Gehrig, S., Pock, T., Bischof, H.: Pushing the limits of stereo using variational stereo estimation. In: Intelligent Vehicles Symposium (IV), 2012 IEEE. pp. 401–407 (June 2012)

[30] Rosen, D.M., Kaess, M., Leonard, J.J.: An incremental trust-region method for robust online sparse least-squares estimation. In: Robotics and Automation (ICRA), 2012 IEEE International Conference on. pp. 1262–1269 (May 2012)

[31] Steinbruecker, F., Sturm, J., Cremers, D.: Real-time visual odometry from dense rgb-d images. In: Workshop on Live Dense Reconstruction with Moving Cameras at the Intl. Conf. on Computer Vision (ICCV) (2011)

[32] Umeyama, S.: Least-squares estimation of transformation parameters between two point patterns. IEEE Trans. Pattern Anal. Mach. Intell. 13(4), 376–380 (Apr 1991)

[33] USGS: The national map download (2015), http://viewer.nationalmap.gov/basic/

[34] Zach, C., Pock, T., Bischof, H.: A duality based approach for realtime tv-l1 optical flow. In: Proceedings of the 29th DAGM Conference on Pattern Recognition. pp. 214–223. Springer-Verlag, Berlin, Heidelberg (2007)