Supplementary Material for ECCV 2016 Paper: TGV-SLAM: A Bayesian Approach to Real-Time Dense Monocular SLAM

Anonymous ECCV submission

Paper ID 274

Abstract. This document accompanies the paper: TGV-SLAM A Bayesian Approach to Real-Time Dense Monocular SLAM. This information is not necessary to understand the paper. However, for those wishing more detailed step-by-step explanations, we provide additional details and intuitions behind some of the derivations here.

A Derivations

A.1 Additional Details of Robust Weight Function

In this section we provide additional justification for our choice of robust weight-inf function. Denoting the photometric residual by δ, we expect inliers to have a (normal) PDF proportional to

$$p_i(\delta) = \exp(-\delta^2), \tag{1}$$

and occluded pixels to have photometric errors distributed uniformly throughout the entire photometric range,

$$p_o(\delta) = \begin{cases} \frac{1}{b-a}, & \delta \in (a,b) \\ 0, & \text{o/w.} \end{cases} \tag{2}$$

Thus, given some mixing parameter $m \in (0,1)$ describing the fraction of outliers, the PDF of the mixture for $\delta \in (a,b)$ is given by

$$p_{mix}(\delta) = m \exp(-\delta^2) + (1-m)\left(1/(b-a)\right). \tag{3}$$

Under this non-normal distribution, the unbiased corrected ML solution is found using iteratively re-weighted linear least squares to minimize

$$\sum_{\mathbf{x}\in\Omega} (I^t(W(\mathbf{x}, Z, \mathbf{p})) - I^0(\mathbf{x}))^2 w(\delta), \tag{4}$$

where the weighting function is given by $w(\delta) = p_i^{-1}(p_{mix}(\delta))$, expanding to

$$w(\delta) = \sqrt{-\ln\left(m \exp(-\delta^2) + (1-m)\left(1/(b-a)\right)\right)}/|\delta|, \tag{5}$$

Shown in Fig. 1, this weighting scheme plateaus with $w(\delta) = 1$ for most 'middling' values, and begins decreasing rapidly beyond a crossover point that occurs around $\delta^2 = -\ln((1-m)/(b-a))$. It also has the undesirable property that $\lim_{\delta \to 0} w(\delta) = \infty$, which could destabilize the solution by essentially putting all the weight on a few inlier values that have exceptionally low error. Therefore, we avoid the instability by instead using the Blake-Zisserman weighting function [7, §A6.8],

$$w_{BZ}(\delta) = \begin{cases} 1, & |\delta| < \tau \\ \tau/|\delta|, & \text{o/w}, \end{cases} \tag{6}$$

where τ may be determined from the mixing ratio as

$$\tau = \sqrt{-\ln((1-m)/(b-a))}. \tag{7}$$

This closely approximates the unbiased weight function in (5), while removing the unstable singularity at $\delta = 0$ (Fig. 1).

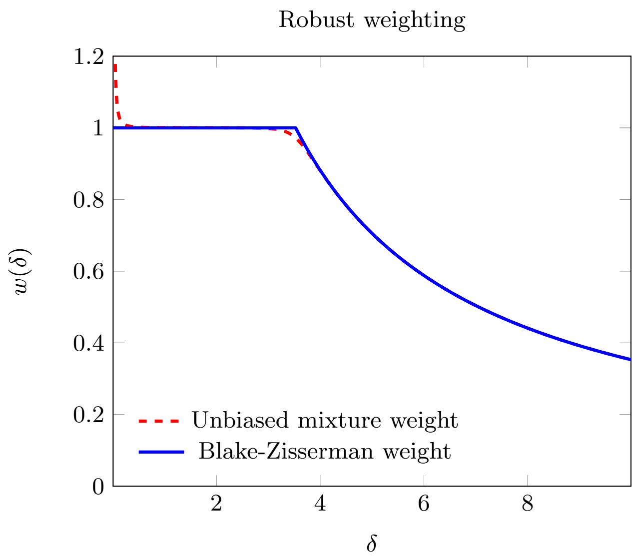

Fig. 1. The weighting suggested by a mixture distribution with uniform outliers goes to infinity for small δ. The Blake-Zisserman weight function is a close approximation that avoids this singularity.

A.2 Understanding the Total Generalized Variation Prior

Without making particular assumptions about *what* is being observed by the camera, the least limiting prior is to assume some level of continunity in the inverse depth map, u. For example, if one assumes that the differences between adjacent inverse depth values should be normally distributed (e.g., the map should be approximately *smooth*), this implies minimizing the squared gradient magnitude,

$$\int_{\Omega} |\nabla u|^2. \tag{8}$$

However, such a regularizer would heavily penalize outliers (e.g., sharp discontinuities), causing blending between foreground and background objects, and eliminating small details in the map.

Much more permissive of discontinuitives is the Total Variation (TV) regularizer,

$$TV(u) = \int_{\Omega} |\nabla u|, \tag{9}$$

which amounts to a prior that the depth should be approximately *piecewise constant*. Unfortunately, this causes undesired flattening as well as stair-stepping where there should be smooth gradients.

To get the smoothness benefits of (8) along with the capacity for discontinuities from (9), but without flattening or stair-stepping, we would prefer a prior that implicitly assumes that inverse depth map is *piecewise planar*. If u is piecewise planar, then ∇u should be piecewise constant; thus, a piecewise planar prior on u may be achieved by applying the TV regularizer onto the components of an auxiliary field $w = [w_x, w_y]^\mathsf{T}$ subject to the constraint $\nabla u = w$. Encoding the latter as a soft constraint with relative weighting α would yield

$$\min_{w} \left\{ \int_{\Omega} \alpha |\nabla u - w| + \int_{\Omega} |\nabla w_x| + \int_{\Omega} |\nabla w_y| \right\}. \tag{10}$$

However, the latter two terms are biased by the orientation of the sampling grid, so that planes having horizontally or vertically aligned slopes would be given higher likelihood. This bias may be corrected by using the 2-dimensional infinitesimal strain tensor [8], a differential operator defined as

$$\mathcal{E}(w) = \frac{1}{2}\left((\nabla w)^\mathsf{T} + \nabla w\right) \tag{11}$$

$$= \begin{bmatrix} \frac{\partial w_x}{\partial x} & \frac{1}{2}\left(\frac{\partial w_x}{\partial y} + \frac{\partial w_y}{\partial x}\right) \\ \frac{1}{2}\left(\frac{\partial w_x}{\partial y} + \frac{\partial w_y}{\partial x}\right) & \frac{\partial w_y}{\partial y} \end{bmatrix}. \tag{12}$$

Technically, the infinitessimal strain tensor is a linearized approximation of the Lagrangian finite strain tensor, which is a good approximation when the displacement gradients are small – but for our purposes, $\mathcal{E}$ may be thought of as simply a generalization of the gradient operator to higher dimensions. For this reason it is sometimes referred to as the *symmetrized gradient* [2, §3.1]).

Replacing the latter two terms in (10) with $\mathcal{E}$, we obtain the Total Generalized Variation (TGV) regularizer of order $k = 2$,

$$TGV_\alpha^2(u) = \min_{w\in\mathbb{R}^2} \left\{ \alpha \int_\Omega |\nabla u - w| \, dx + \int_\Omega |\mathcal{E}(w)| \, dx \right\}, \tag{13}$$

Like the TV prior, the TGV prior admits discontinuities, but whereas the TV prior has a tendency to flatten out regions, the TGV prior merely has a tendency to break smooth surfaces into planar patches – an effect that is often unnoticed by the human eye, and produces visually smooth and pleasing results. In addition, it is excellent at aggregating pixels that are part of planar regions, such as man-made surfaces, or ground surfaces when viewed from high elevation.

A.3 Additional Details of 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,

$$\hat{\mathbf{p}}^t_{ML} = \operatorname{argmin}_{\mathbf{p}} E(Z^0, \mathbf{p}^t) \tag{14}$$

$$= \operatorname{argmin}_{\mathbf{p}} \sum_{\mathbf{x}\in\Omega} (I^t(W(\mathbf{x}, Z^0, \mathbf{p}^t)) - I^0(\mathbf{x}))^2 w(\delta). \tag{15}$$

For notational simplicity, we drop the t superscripts and denote the template image I^0 as T. Ignoring the weighting term, the problem is to solve

$$\operatorname{argmin}_{\mathbf{p}} \sum_{\mathbf{x}\in\Omega} (I(W(\mathbf{x}; \mathbf{p})) - T(\mathbf{x}))^2. \tag{16}$$

Using the forward-additive approach [1], we want to find the pose parameters $\mathbf{p}$ that define a mapping from pixel coordinates in the reference image (previous pose) to pixel coordinates in the template image (new pose). Rather than directly estimating $\mathbf{p}$, we start from an initial estimate and then iteratively solve for the update vector $\Delta\mathbf{p}$. We use the $\oplus$ operator to denote application of the update vector (e.g., $\mathbf{p} \leftarrow \mathbf{p} \oplus \Delta\mathbf{p}$). Thus, on each iteration our objective is to solve

$$\operatorname{argmin}_{\Delta\mathbf{p}} \sum_{\mathbf{x}\in\Omega} (I(W(\mathbf{x}; \mathbf{p} \oplus \Delta\mathbf{p})) - T(\mathbf{x}))^2. \tag{17}$$

The above is linearized using a first-order Taylor series expansion to give

$$\sum_{\mathbf{x}} (I(W(\mathbf{x};\mathbf{p})) + \mathbf{J}(\mathbf{x})\Delta\mathbf{p} - T(\mathbf{x}))^2, \tag{18}$$

where $\mathbf{J}(\mathbf{x})$ is the 1×6 Jacobian of $I(W(\mathbf{x};\mathbf{p}))$. By the chain rule,

$$\mathbf{J}(\mathbf{x}) = \nabla I^{\mathsf{T}} \frac{\partial W(\mathbf{x};\mathbf{p})}{\partial \mathbf{p}}, \tag{19}$$

where $\nabla I = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right)^{\mathsf{T}}$ is the gradient of I evaluated at $W(\mathbf{x};\mathbf{p})$ and $\frac{\partial W(\mathbf{x};\mathbf{p})}{\partial \mathbf{p}}$ is the 2×6 Jacobian of $W(\mathbf{x};\mathbf{p})$.

Taking the partial derivative of (18) with respect to $\Delta\mathbf{p}$ and setting equal to zero (to find the minimum, assuming we are in a local basin of attraction) gives

$$\frac{\partial}{\partial \Delta\mathbf{p}} \sum_{\mathbf{x}} (I(W(\mathbf{x};\mathbf{p})) + \mathbf{J}(\mathbf{x})\Delta\mathbf{p} - T(\mathbf{x}))^2 \tag{20}$$

$$\sum_{\mathbf{x}} \mathbf{J}(\mathbf{x})^{\mathsf{T}} (I(W(\mathbf{x};\mathbf{p})) + \mathbf{J}(\mathbf{x})\Delta\mathbf{p} - T(\mathbf{x})) = 0. \tag{21}$$

With some rearrangement, we obtain a 6-dimensional linear system to solve for $\Delta\mathbf{p}$,

$$\sum_{\mathbf{x}} \mathbf{J}(\mathbf{x})^{\mathsf{T}} \mathbf{J}(\mathbf{x}) \Delta\mathbf{p} = \sum_{\mathbf{x}} \mathbf{J}(\mathbf{x})^{\mathsf{T}} E(\mathbf{x}), \tag{22}$$

where

$$E(\mathbf{x}) = T(\mathbf{x}) - I(W(\mathbf{x};\mathbf{p})) \tag{23}$$

is the photometric error.

Iterative application of (22) leads to a solution via the Gauss-Newton method. Standard enhancements such as Levenberg-Marquardt or Powell's dog leg may also be applied, but we find a simple Gauss-Newton method with line search in the update direction to be most efficient.

To improve the basin of convergence, we re-solve the problem in a coarse to fine approach using an image pyramid.

For implementation efficiency, we compute all the residuals and partial derivatives in parallel using CUDA, and perform the final summation on the GPU using a reduction kernel.

A.4 Additional Details of Map Update Step

In the depth map update step, we compute the maximum *a posteriori* inverse depth map under a fixed estimate of relative pose,

$$\hat{Z}^0_{MAP} = \text{argmin}_Z\, E(Z, \mathbf{p}^t) \tag{24}$$
$$= TGV^2_\alpha(u) + \lambda F(u). \tag{25}$$

This is accomplished using the primal dual method [3], a recently developed variational method that can efficiently solve non-differentiable saddle point problems of the form

$$\min_{x\in\mathcal{X}} \max_{y\in\mathcal{Y}} \langle \mathcal{K}x, y\rangle + \mathcal{F}(x) - \mathcal{G}(y), \tag{26}$$

where $\mathcal{K}$ is a linear operator, and $\mathcal{F}$ and $\mathcal{G}$ are arbitrary functions.

The algorithm is based on the *resolvent* operators for the subgradients of $\mathcal{F}$ and $\mathcal{G}$, denoted by $(\mathtt{I} + \tau\partial\mathcal{F})^{-1}$ and $(\mathtt{I} + \sigma\partial\mathcal{G})^{-1}$ respectively, and defined by

$$(\mathtt{I} + \tau\partial\mathcal{F})^{-1}(\bar{x}) = \text{argmin}_x \frac{||x-\bar{x}||^2}{2} + \tau\mathcal{F}(x). \tag{27}$$

From this definition, the resolvent operator may be interpreted as finding a solution (x) that minimizes $\mathcal{F}(x)$ in the vicinity (or proximity) of an existing initial guess $\bar{x}$, where τ controls the weighting on $\mathcal{F}$. For this reason, it is often referred to as the *proximal* mapping and denoted $\text{prox}^\tau_\mathcal{F}(\bar{x})$.

Given some initial values for x and y and step sizes $\sigma > 0$, $\tau > 0$ the primal-dual method is given by

Algorithm 1 Generic primal dual method

1: $x \leftarrow$ initial estimate
2: **repeat**
3: $\quad y \leftarrow \text{prox}^\sigma_\mathcal{G}(y + \sigma\mathcal{K}x)$
4: $\quad \bar{x} \leftarrow \text{prox}^\tau_\mathcal{F}(x - \tau\mathcal{K}^\dagger y)$
5: $\quad x \leftarrow 2\bar{x} - x$
6: **until** converged

This method has been proven to converge if $\mathcal{F}$ and $\mathcal{G}$ are both convex functions, and $\sigma\tau\,||\mathcal{K}||^2 < 1$, where $||\mathcal{K}||$ is the operator norm of $\mathcal{K}$ [3].

We now show how (25) may be cast in the form of (26) so that it may be optimized using Algorithm 1. First, recall that the primal form of the TGV regularizer is given in (13) by

$$TGV_\alpha^2(u) = \min_{w \in \mathbb{R}^2} \left\{ \alpha_1 \int_\Omega |\nabla u - w| dx + \alpha_0 \int_\Omega |\mathcal{E}(w)| dx \right\}. \tag{28}$$

Thus, any arbitrary TGV-regularized problem may be written in discrete form as

$$\min_{u,w} \{F(u) + \alpha_1 |\nabla u - w| + \alpha_0 |\mathcal{E}(w)|\}, \tag{29}$$

where $F(u)$ is the data term.

Let us define the scaled indicator function as

$$\mathcal{I}_k(x) = \begin{cases} 0, & |x| \leq k \\ \infty, & o/w. \end{cases} \tag{30}$$

Taking the Legendre-Fenchel transform of (30), we obtain an optimization problem that is solved by the scaled L^2-norm,

$$\alpha |x| = \max_p \{\langle p, x \rangle - \mathcal{I}_\alpha(p)\}. \tag{31}$$

Thus, we may use (31) to replace the scaled vector norms in (29) to obtain a saddle point optimization problem in two additional dual variables p, q:

$$\min_{u,w} \{\max_{p,q} \{F(u) + \langle \nabla u - w, p \rangle - \mathcal{I}_{\alpha_1}(p) + \langle \mathcal{E}(w), q \rangle - \mathcal{I}_{\alpha_0}(q)\}\} \tag{32}$$

where p is the dual of $\nabla u - w$, and q is the dual of $\mathcal{E}(w)$.

By combining the two inner products into one, (32) can be rewritten as

$$\min_{u,w} \max_{p,q} \left\{ \left\langle \begin{bmatrix} \nabla & -\mathtt{I} \\ 0 & \mathcal{E} \end{bmatrix} \begin{bmatrix} u \\ w \end{bmatrix}, \begin{bmatrix} p \\ q \end{bmatrix} \right\rangle + \right. \tag{33}$$

$$\left. F(u) - \mathcal{I}_{\alpha_0}(q) - \mathcal{I}_{\alpha_1}(p) \right\}. \tag{34}$$

If we let $\mathcal{X} = U \times W$ and $\mathcal{Y} = P \times Q$, then it becomes clear that our problem is in the form of (26), where

$$\mathcal{F}(x) = \lambda F(u) \tag{35}$$

$$\mathcal{G}(y) = \mathcal{I}_{\alpha_0}(p) + \mathcal{I}_{\alpha_1}(q) \tag{36}$$

$$\mathcal{K} = \begin{bmatrix} \nabla & -\mathtt{I} \\ 0 & \mathcal{E} \end{bmatrix}. \tag{37}$$

Let us now write line 3 of Alg. 1 in terms of the expanded variables. Because $y = (u, w)$, this results in two independent equations,

$$p \leftarrow \text{prox}^{\sigma}_{\mathcal{I}_{\alpha_1}}(p + \sigma(\nabla u - w)) \tag{38}$$
$$q \leftarrow \text{prox}^{\sigma}_{\mathcal{I}_{\alpha_0}}(q + \sigma\mathcal{E}(w)). \tag{39}$$

We can further simplify the proximal mappings of the scaled indicator function. From the definition in (27) we have

$$\text{prox}^{\sigma}_{\mathcal{I}_{\alpha_1}}(\bar{p}) = \text{argmin}_p \frac{||p - \bar{p}||^2}{2} + \sigma\mathcal{I}_{\alpha_1}(p). \tag{40}$$

Clearly, the minimum is achieved by avoiding the infinite case in the indicator function, which means finding $||p|| \leq \alpha_1$. The other term requires we find p closest to $\bar{p}$. Therefore, the solution is simply to project $\bar{p}$ down to the sphere of radius α_1. Thus,

$$\text{prox}^{\sigma}_{\mathcal{I}_{\alpha_1}}(\bar{p}) = \begin{cases} \bar{p} & ||\bar{p}|| < \alpha_1 \\ \alpha_1 \frac{\bar{p}}{||\bar{p}||} & o/w \end{cases} \tag{41}$$
$$= \frac{\bar{p}}{\max(1, ||\bar{p}||/\alpha_1)} = \text{proj}_{\alpha_1}(\bar{p}). \tag{42}$$

We must also write line 4 of Alg. 1 using the expanded variables, and again because $x = (u, w)$ this will result in two more update equations, but first we need the adjoint of $\mathcal{K}^\dagger$, which may be written blockwise as

$$\mathcal{K}^\dagger = \begin{bmatrix} \nabla^\dagger & 0 \\ -\mathtt{I} & \mathcal{E}^\dagger \end{bmatrix}. \tag{43}$$

Substituting in, we obtain

$$u \leftarrow \text{prox}^{\tau}_{F}(u - \tau\nabla^\dagger p) \tag{44}$$
$$w \leftarrow \text{prox}^{\tau}_{F}(w - \tau(-p + \mathcal{E}^\dagger q)). \tag{45}$$

Because $\mathcal{F}(x) = \lambda F(u)$ is independent of w, the proximal mapping in (45) reduces to identity,

$$\text{prox}^{\tau}_{\mathcal{F}}(\bar{w}) = \text{argmin}_w \frac{||w - \bar{w}||^2}{2} = \bar{w}, \tag{46}$$

and hence

$$w \leftarrow w + \tau(p - \mathcal{E}^{\dagger} q). \tag{47}$$

The proximal mapping in (44) depends on the data fidelity defined in (**??**) evaluated per-pixel and weighted by λ, written as

$$F(u) = \frac{\lambda(I(W(\mathbf{x}, u)) - T(x))^2}{2}, \tag{48}$$

where we drop the implicit dependence on pose $\mathbf{p}$ in the warping function, and u is the pixel's inverse depth.

In order to compute the proximal mapping we need a closed form solution for $F(u)$. To this end we locally approximate $I(W(u|\mathbf{x}))$ by a linearization around $\bar{u}$,

$$I(W(u|\mathbf{x})) \approx I(W(\bar{u}|\mathbf{x})) + (u - \bar{u})\frac{\partial I(W(\bar{u}|\mathbf{x}))}{\partial u} \tag{49}$$

where

$$\frac{\partial I(W(\bar{u}|\mathbf{x}))}{\partial u} = \nabla I(W(\bar{u}|\mathbf{x}))^{\mathsf{T}} \frac{\partial W(\bar{u}|\mathbf{x})}{\partial u}. \tag{50}$$

Substituting in, and making some notational simplifications, we obtain the following closed form local approximation:

$$F(u) \approx \frac{\lambda(I(W(\bar{u}|\mathbf{x})) + (u - \bar{u})\frac{\partial I(W(\bar{u}|\mathbf{x}))}{\partial u} - T(\mathbf{x}))^2}{2} \tag{51}$$

$$\approx \frac{\lambda(I(\mathbf{x}') + (u - \bar{u})\frac{\partial I}{\partial u} - T(\mathbf{x}))^2}{2}. \tag{52}$$

Substituting into (27), we obtain

$$\mathrm{prox}^{\tau}_{\mathcal{F}}(\bar{u}) = \mathrm{argmin}_u \frac{(u - \bar{u})^2}{2} + \frac{\tau\lambda(I(\mathbf{x}') + (u - \bar{u})\frac{\partial I}{\partial u} - T(\mathbf{x}))^2}{2}. \tag{53}$$

To find the minimum, we take the derivative and equate it to zero in order to solve for u. Defining $a = \frac{\partial I}{\partial u}$ and $b = I(\mathbf{x}') - \bar{u}\frac{\partial I}{\partial u} - T(\mathbf{x})$, this gives

$$0 = \frac{2(u - \bar{u})}{2} + \frac{\tau\lambda(2ua^2 + 2ab)}{2} \tag{54}$$

$$= (u - \bar{u}) + \tau\lambda(ua^2 + ab) \tag{55}$$

Solving for u, we find

$$\text{prox}^{\tau}_{\mathcal{F}}(\bar{u}) = \frac{\bar{u} - \tau\lambda ab}{1 + \tau\lambda a^2}. \tag{56}$$

Finally, taking the results of (38) (39), (42), (44), (47), (56) and substituting into Algorithm 1 we obtain the primal dual algorithm for regularized inverse depth estimation:

Algorithm 2 Regularized inverse depth map estimation

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

The first two steps (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.

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] 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
[4] 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)
[5] Engel, J., Schöps, T., Cremers, D.: Lsd-slam: Large-scale direct monocular slam. In: Computer Vision–ECCV 2014, pp. 834–849. Springer (2014)
[6] Engel, Jakob: Lsd slam source code (2015), https://github.com/tum-vision/lsd_slam

[7] Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, New York, NY, USA, 2 edn. (2003)
[8] Wikipedia: Infinitesimal strain tensor — Wikipedia, the free encyclopedia (2015), `http://en.wikipedia.org/wiki/Infinitesimal_strain_theory#Infinitesimal_strain_tensor`