Stochastic Variational Inference with Gradient Linearization
– Supplemental Material –

Tobias Plötz* Anne S. Wannenwetsch* Stefan Roth
Department of Computer Science, TU Darmstadt

Preface. In this supplemental material, we show that SVIGL can be interpreted as a gradient descent approach using a special preconditioner and provide the gradient linearization for the optical flow and Poisson-Gaussian energies used in the main manuscript. Furthermore, we provide a proof for Proposition 2, show the hyperparameter evaluation for SVI with SGD, and give additional details of our optical flow experiment in Table 2. Finally, we show some exemplary results of Poisson-Gaussian denoising and provide details on the experiment on 3D surface reconstruction.

A. SVIGL as Preconditioned Gradient Descent

Here, we show that an update step of SVIGL as given in Eq. (12) can be interpreted as one iteration of preconditioned gradient descent. To simplify notation let $\mathbf{A}_{\boldsymbol{\theta}} \equiv \mathbf{A}_{\boldsymbol{\theta}}\big(\boldsymbol{\theta}^{(t)}\big)$ and $\mathbf{b}_{\boldsymbol{\theta}} \equiv \mathbf{b}_{\boldsymbol{\theta}}\big(\boldsymbol{\theta}^{(t)}\big)$. Following, *e.g.* [29], we have

$$\begin{aligned}
\boldsymbol{\theta}^{(t+1)} &= -\mathbf{A}_{\boldsymbol{\theta}}^{-1}\mathbf{b}_{\boldsymbol{\theta}} && \text{(20a)}\\
&= \boldsymbol{\theta}^{(t)} - \mathbf{A}_{\boldsymbol{\theta}}^{-1}\mathbf{b}_{\boldsymbol{\theta}} - \boldsymbol{\theta}^{(t)} && \text{(20b)}\\
&= \boldsymbol{\theta}^{(t)} - \mathbf{A}_{\boldsymbol{\theta}}^{-1}\Big(\mathbf{b}_{\boldsymbol{\theta}} + \mathbf{A}_{\boldsymbol{\theta}}\boldsymbol{\theta}^{(t)}\Big) && \text{(20c)}\\
&= \boldsymbol{\theta}^{(t)} - \mathbf{A}_{\boldsymbol{\theta}}^{-1}\nabla_{\boldsymbol{\theta}}\,\mathrm{KL}\,(q\,||\,p). && \text{(20d)}
\end{aligned}$$

Therefore, SVIGL performs gradient descent with preconditioner $P = \mathbf{A}_{\boldsymbol{\theta}}^{-1}$. This interpretation also allows to introduce a step size parameter α to SVIGL

$$\begin{aligned}
\boldsymbol{\theta}^{(t+1)} &= \boldsymbol{\theta}^{(t)} - \alpha\mathbf{A}_{\boldsymbol{\theta}}^{-1}\nabla_{\boldsymbol{\theta}}\,\mathrm{KL}\,(q\,||\,p) && \text{(21a)}\\
&= \boldsymbol{\theta}^{(t)} - \alpha\mathbf{A}_{\boldsymbol{\theta}}^{-1}\Big(\mathbf{b}_{\boldsymbol{\theta}} + \mathbf{A}_{\boldsymbol{\theta}}\boldsymbol{\theta}^{(t)}\Big) && \text{(21b)}\\
&= (1-\alpha)\boldsymbol{\theta}^{(t)} + \alpha\hat{\boldsymbol{\theta}}^{(t+1)}, && \text{(21c)}
\end{aligned}$$

with $\hat{\boldsymbol{\theta}}^{(t+1)} = -\mathbf{A}_{\boldsymbol{\theta}}^{-1}\mathbf{b}_{\boldsymbol{\theta}}$ denoting the SVIGL estimate as given in Eq. (12). In practice, our experiments have shown that the performance of SVIGL is not sensitive to the choice of the step size parameter. We thus simply set $\alpha = 1$.

*Authors contributed equally

B. Linearized Gradients

In the following, we show how linearized gradients can be obtained for the presented applications of SVIGL in optical flow estimation and Poisson-Gaussian denoising. For other applications, including many models in computer vision, it is possible to derive parameters $\mathbf{A}_{\boldsymbol{\theta}}$ and $\mathbf{b}_{\boldsymbol{\theta}}$ in a similar fashion.

B.1. Optical flow

Here, we show the derivation of a linearized gradient for a simple optical flow energy using the brightness constancy assumption, *i.e.*

$$\begin{aligned}
E(\mathbf{x},\mathbf{y}) =& \lambda_{\mathrm{D}}\sum_{l=1}^{L}\rho_{\mathrm{D}}\left(I_{t,l} + \begin{pmatrix} I_{x,l}\\ I_{y,l}\end{pmatrix}^{\mathrm{T}}\left(\mathbf{x}_l - \mathbf{x}_l^0\right)\right)\\
&+ \lambda_{\mathrm{S}}\sum_{j=1}^{J}\sum_{l=1}^{L}\rho_{\mathrm{S}}\left(\left\|\left(\mathbf{f}_j * \mathbf{x}\right)_l\right\|_2\right) && \text{(22a)}\\
=& \lambda_{\mathrm{D}}E_{\mathrm{D}}(\mathbf{x},\mathbf{y}) + \lambda_{\mathrm{S}}E_{\mathrm{S}}(\mathbf{x}), && \text{(22b)}
\end{aligned}$$

with $I_{t,l} = I_2\left(l + \mathbf{x}_l^0\right) - I_1\left(l\right)$, $\begin{pmatrix} I_{x,l}\\ I_{y,l}\end{pmatrix} = \nabla I_2\left(l + \mathbf{x}_l^0\right)$, and $\mathbf{x}_l^0$ denoting the point of approximation of the Taylor linearization. The derivations for the EpicFlow energy function in Eq. (15) are more tedious, but can be done analogously.

Data term. In a first step, we derive the linearized gradient for the data energy term. Here, it holds that

$$\begin{aligned}
\nabla_{\mathbf{x}_l}E_{\mathrm{D}}(\mathbf{x},\mathbf{y}) =& \nabla_{\mathbf{x}_l}\rho_{\mathrm{D}}\left(I_{t,l} + \begin{pmatrix} I_{x,l}\\ I_{y,l}\end{pmatrix}^{\mathrm{T}}\left(\mathbf{x}_l - \mathbf{x}_l^0\right)\right) && \text{(23a)}\\
=& \rho_{\mathrm{D}}'\left(I_{t,l} + \begin{pmatrix} I_{x,l}\\ I_{y,l}\end{pmatrix}^{\mathrm{T}}\left(\mathbf{x}_l - \mathbf{x}_l^0\right)\right)\cdot\begin{pmatrix} I_{x,l}\\ I_{y,l}\end{pmatrix}. && \text{(23b)}
\end{aligned}$$

The derivative of the generalized Charbonnier [2] used for

$\rho_{\text{D}}(\cdot)$ can be written as:

$$\rho'_{\text{D}}(x) = \frac{x}{c^2}\left(\frac{(x/c)^2}{\max(1, 2-a)} + 1\right)^{(a/2-1)} \tag{24a}$$
$$\equiv \tilde{\rho}_{\text{D}}(x)\ x. \tag{24b}$$

Using Eqs. (23b) and (24b), we have

$$\nabla_{\mathbf{x}_l} E_{\text{D}}(\mathbf{x}, \mathbf{y}) =$$
$$= \tilde{\rho}_{\text{D}}\left(I_{t,l} + \begin{pmatrix} I_{x,l} \\ I_{y,l} \end{pmatrix}^{\text{T}} \left(\mathbf{x}_l - \mathbf{x}_l^0\right)\right)$$
$$\cdot \left(\begin{pmatrix} I_{x,l}I_{t,l} \\ I_{y,l}I_{t,l} \end{pmatrix} + \begin{pmatrix} I_{x,l}^2 & I_{x,l}I_{y,l} \\ I_{x,l}I_{y,l} & I_{y,l}^2 \end{pmatrix}\left(\mathbf{x}_l - \mathbf{x}_l^0\right)\right). \tag{25}$$

The last identity (Eq. 25) allows us to easily identify a linearized form of the gradient of the data term as

$$\nabla_{\mathbf{x}} E_{\text{D}}(\mathbf{x}, \mathbf{y}) = \mathbf{A}_{\mathbf{x}}^{\text{D}}(\mathbf{x})\mathbf{x} + \mathbf{b}_{\mathbf{x}}^{\text{D}}(\mathbf{x}), \tag{26}$$

with

$$\mathbf{A}_{\mathbf{x}}^{\text{D}}(\mathbf{x}) = \begin{pmatrix} \mathbf{D}\left(\tilde{\rho}_{\text{D}} \cdot I_x^2\right) & \mathbf{D}\left(\tilde{\rho}_{\text{D}} \cdot I_x I_y\right) \\ \mathbf{D}\left(\tilde{\rho}_{\text{D}} \cdot I_x I_y\right) & \mathbf{D}\left(\tilde{\rho}_{\text{D}} \cdot I_y^2\right) \end{pmatrix} \tag{27}$$

and

$$\mathbf{b}_{\mathbf{x}}^{\text{D}}(\mathbf{x}) = \begin{pmatrix} \mathbf{D}\left(\tilde{\rho}_{\text{D}} \cdot I_x I_t\right)\mathbf{1} \\ \mathbf{D}\left(\tilde{\rho}_{\text{D}} \cdot I_y I_t\right)\mathbf{1} \end{pmatrix} - \mathbf{A}_{\mathbf{x}}(\mathbf{x})\mathbf{x}^0. \tag{28}$$

Here, $\mathbf{x} = \left(x_1^{(1)}, \ldots, x_L^{(1)}, x_1^{(2)}, \ldots, x_L^{(2)}\right)^{\text{T}}$ denotes the stacked vector of all horizontal and vertical flow components. $\mathbf{D}(\cdot)$ turns the argument vector into a diagonal matrix (short for $\text{diag}\{\cdot\}$), and the product is applied element-wise.

Smoothness term. For the smoothness term let us first express the convolution $\mathbf{f}_j * \mathbf{x}$ as a matrix-vector product $\mathbf{F}_j \cdot \mathbf{x}$, with $\mathbf{F}_j$ denoting the convolution matrix corresponding to $\mathbf{f}_j$ and $\mathbf{x}$ the vectorized flow as before. With that, the gradient of the smoothness term E_{S} can be written as:

$$\nabla_{\mathbf{x}} E_{\text{S}}(\mathbf{x}) = \nabla_{\mathbf{x}} \sum_{j=1}^{J}\sum_{l=1}^{L} \rho_{\text{S}}\left(\left(\mathbf{F}_j \mathbf{x}\right)_l\right) \tag{29a}$$
$$= \sum_{j=1}^{J} \mathbf{F}_j^T \rho'_{\text{S}}(\mathbf{F}_j \mathbf{x}). \tag{29b}$$

Using the derivative ρ'_{S} as given in Eq. (24b), we obtain

$$\sum_{j=1}^{J} \mathbf{F}_j^T \rho'_{\text{S}}(\mathbf{F}_j \mathbf{x}) = \sum_{j=1}^{J} \mathbf{F}_j^T\ \mathbf{D}\Big(\tilde{\rho}_{\text{S}}(\mathbf{F}_j \mathbf{x})\Big)\ \mathbf{F}_j \mathbf{x} \tag{30a}$$
$$= \left(\sum_{j=1}^{J} \mathbf{F}_j^T\ \mathbf{D}\Big(\tilde{\rho}_{\text{S}}(\mathbf{F}_j \mathbf{x})\Big)\ \mathbf{F}_j\right)\mathbf{x} \tag{30b}$$
$$\equiv \mathbf{A}_{\mathbf{x}}^{\text{S}}(\mathbf{x})\mathbf{x}. \tag{30c}$$

Complete linearized gradient. We now summarize the results of Eqs. (27), (28), and (30c) to obtain the linearized gradient as

$$\nabla_{\mathbf{x}} E(\mathbf{x}, \mathbf{y}) = \lambda_{\text{D}} \nabla_{\mathbf{x}} E_{\text{D}}(\mathbf{x}, \mathbf{y}) + \lambda_{\text{S}} \nabla_{\mathbf{x}} E_{\text{S}}(\mathbf{x}) \tag{31a}$$
$$= \Big(\lambda_{\text{D}} \mathbf{A}_{\mathbf{x}}^{\text{D}}(\mathbf{x}) + \lambda_{\text{S}} \mathbf{A}_{\mathbf{x}}^{\text{S}}(\mathbf{x})\Big)\mathbf{x} + \lambda_{\text{D}} \mathbf{b}_{\mathbf{x}}^{\text{D}} \tag{31b}$$
$$\equiv \mathbf{A}_{\mathbf{x}}(\mathbf{x})\mathbf{x} + \mathbf{b}_{\mathbf{x}}. \tag{31c}$$

B.2. Poisson-Gaussian denoising

Let us first recap the energy function for Poisson-Gaussian denoising:

$$E(\mathbf{x}, \mathbf{y}) = \frac{\lambda_{\text{D}}}{2} \sum_{l=1}^{L} \frac{(\mathbf{x}_l - \mathbf{y}_l)^2}{\sigma(\mathbf{x}_l)^2} \tag{32a}$$
$$+ \lambda_{\text{S}} \sum_{j=1}^{J}\sum_{l=1}^{L} \rho_{\text{S}}\Big(\left(\mathbf{f}_j * \mathbf{x}\right)_l\Big),$$
$$= \lambda_{\text{D}} E_{\text{D}}(\mathbf{x}, \mathbf{y}) + \lambda_{\text{S}} E_{\text{S}}(\mathbf{x}), \tag{32b}$$

where

$$\sigma(\mathbf{x}_l)^2 = \beta_1 \mathbf{x}_l + \beta_2. \tag{33}$$

We will derive the linearized gradients for the data term E_{D} and the smoothness term E_{S} separately.

Data term. The gradient of the data term is given as

$$\nabla_{\mathbf{x}} E_{\text{D}}(\mathbf{x}, \mathbf{y})$$
$$= \frac{(\mathbf{x} - \mathbf{y})}{\sigma(\mathbf{x})^2} - \frac{\beta_1(\mathbf{x} - \mathbf{y})^2}{2\sigma(\mathbf{x})^4} \tag{34a}$$
$$= \frac{\mathbf{x}}{\sigma(\mathbf{x})^2} - \frac{\mathbf{y}}{\sigma(\mathbf{x})^2} - \frac{\beta_1 \mathbf{x}^2}{2\sigma(\mathbf{x})^4} + \frac{\beta_1 \mathbf{x}\mathbf{y}}{\sigma(\mathbf{x})^4} - \frac{\beta_1 \mathbf{y}^2}{2\sigma(\mathbf{x})^4} \tag{34b}$$
$$= \mathbf{x}\left(\frac{1}{\sigma(\mathbf{x})^2} - \frac{\beta_1 \mathbf{x}}{2\sigma(\mathbf{x})^4} + \frac{\beta_1 \mathbf{y}}{\sigma(\mathbf{x})^4}\right)$$
$$- \left(\frac{\mathbf{y}}{\sigma(\mathbf{x})^2} + \frac{\beta_1 \mathbf{y}^2}{2\sigma(\mathbf{x})^4}\right), \tag{34c}$$

where all operations are element-wise. The linearized gradient of the data term can then be obtained as

$$\mathbf{A}_{\mathbf{x}}^{\mathrm{D}}(\mathbf{x}) = \mathbf{D}\left(\frac{1}{\sigma(\mathbf{x})^2} - \frac{\beta_1 \mathbf{x}}{2\sigma(\mathbf{x})^4} + \frac{\beta_1 \mathbf{y}}{\sigma(\mathbf{x})^4}\right) \tag{35}$$

$$\mathbf{b}_{\mathbf{x}}^{\mathrm{D}}(\mathbf{x}) = -\left(\frac{\mathbf{y}}{\sigma(\mathbf{x})^2} + \frac{\beta_1 \mathbf{y}^2}{2\sigma(\mathbf{x})^4}\right). \tag{36}$$

Smoothness term. For the smoothness term we can reuse the linearized gradient derived in Eq. (30c).

Complete linearized gradient. We can now put the results of Eqs. (30c), (35), and (36) together to obtain a linearized gradient of the energy for Poisson-Gaussian denoising, *c.f*. Eqs. (31a) – (31c).

C. Proof Proposition 2

In this section, we provide a proof for Proposition 2 of the main paper.

Proposition 2. *An energy function can be linearized with a positive semi-definite matrix* $\mathbf{A}_{\mathbf{x}}$ *if it is composed of a sum of energy terms* $\rho_i(\mathbf{w}_i)$ *that fulfill the following conditions:*

1. *Each penalty function* $\rho_i(\cdot)$ *is symmetric and* $\rho_i'(\mathbf{w}_i) \geq 0$ *for all* $\mathbf{w}_i \geq 0$. $(\star)$
2. *Each penalty function* $\rho_i(\cdot)$ *is applied element-wise on* $\mathbf{w}_i$*, which is of the form* $\mathbf{w}_i = \mathbf{K}_i\mathbf{x} + \mathbf{g}_i(\mathbf{y})$*, with filter matrix* $\mathbf{K}_i$ *and* $\mathbf{g}_i$ *not depending on* $\mathbf{x}$. $(\star\star)$

Proof. From assuming a symmetric $\rho_i(\cdot)$ in $(\star)$, it follows that $\rho_i'(\cdot)$ is point symmetric. Due to $\rho_i'(\mathbf{w}_i) \geq 0$ for all $\mathbf{w}_i \geq 0$ we then find that $\rho_i'(\mathbf{w}_i)$ can be written as

$$\rho_i'(\mathbf{w}_i) \equiv \tilde{\rho}_i(\mathbf{w}_i) \cdot \mathbf{w}_i \quad \text{with a} \quad \tilde{\rho}_i(\mathbf{w}_i) \geq 0. \tag{37}$$

For an energy term as described in $(\star\star)$, the gradient w.r.t. $\mathbf{x}$ is given as

$$\nabla_{\mathbf{x}} \rho_i(\mathbf{w}_i) = \mathbf{K}_i^{\mathrm{T}} \cdot \mathbf{C}_i \cdot (\mathbf{K}_i \cdot \mathbf{x} + \mathbf{g}_i(\mathbf{y})), \tag{38}$$

$$\text{with} \quad \mathbf{C}_i = \mathbf{D}\big(\tilde{\rho}_i\left(\mathbf{K}_i \cdot \mathbf{x} + \mathbf{g}_i(\mathbf{y})\right)\big). \tag{39}$$

A linearization can then be obtained using

$$\mathbf{A}_{\mathbf{x}}^i = \mathbf{K}_i^{\mathrm{T}} \cdot \mathbf{C}_i \cdot \mathbf{K}_i, \qquad \mathbf{b}_{\mathbf{x}}^i = \mathbf{K}_i^{\mathrm{T}} \cdot \mathbf{C}_i \cdot \mathbf{g}_i(\mathbf{y}). \tag{40}$$

Since $\mathbf{C}_i$ is a diagonal matrix of non-negative elements (Eq. 37), $\mathbf{A}_{\mathbf{x}}^i$ is positive semi-definite as

$$\mathbf{x}^{\mathrm{T}}\mathbf{A}_{\mathbf{x}}^i\mathbf{x} = \mathbf{x}^{\mathrm{T}}\mathbf{K}_i^{\mathrm{T}}\mathbf{C}_i\mathbf{K}_i\mathbf{x} = \mathbf{v}^{\mathrm{T}}\mathbf{C}_i\mathbf{v} \geq 0. \tag{41}$$

As the sum of positive semi-definite matrices is positive semi-definite, a matrix $\mathbf{A}_{\mathbf{x}}$ composed of energy terms that fulfill $(\star)$ and $(\star\star)$ is positive semi-definite. □

D. Hyperparameters for SGD

In the following, we aim to find optimal hyperparameters for the SVI baseline based on SGD. For all experiments we select an initial step size α_0, which is cut after each third of iterations by a factor of ten. An evaluation of the unnormalized KL divergence for optical flow plotted against the runtime for different initial step sizes α_0 of SGD is shown in Fig. 6a. Here, the KL divergence deteriorates severely using SGD with a step size larger than 10^{-6}. For smaller step sizes, SVI with SGD shows a slow convergence such that we set $\alpha_0 = 10^{-6}$.

Following the same procedure, we perform several experiments for Poisson-Gaussian denoising and evaluate different settings for the initial step size parameter α_0 of SGD in Fig. 7a. Again, an initial step size $\alpha_0 = 10^{-6}$ proves to be most effective. Smaller step sizes converge too slowly, while SGD with bigger step size values converges faster but to a worse local optimum. For an initial step size of $\alpha_0 = 10^{-5}$ optimization diverges immediately.

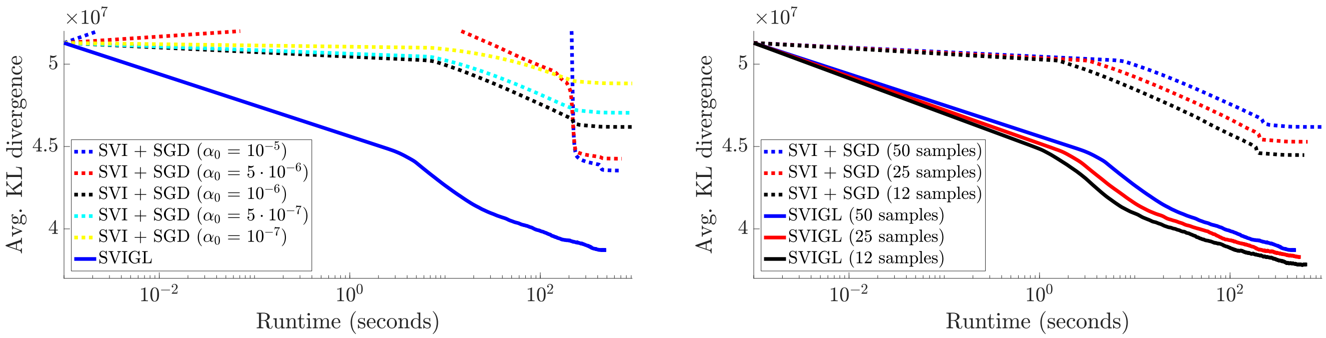
Applying SVI with SGD, we observe in both applications a faster convergence of the KL divergence with a smaller sample size, but a larger number of iterations, *c.f*. Figs. 6b and 7b. We therefore choose $|\mathcal{Z}| = 12$ with 4000 iterations of SGD for the experiments in the main paper.

E. Comparison with ProbFlowFields

In Table 2 of the main paper we evaluate the quality of the posterior variances obtained with SVIGL. Here, we follow Wannenwetsch *et al*. [45] and derive an uncertainty measure by computing the marginal entropy of the flow at every pixel. To have a fair comparison with [45], we use the same EpicFlow [32] energy formulation with learned Gaussian scale mixture penalty functions and explicit indicator variables for their mixture components. Since SVIGL is designed for variational inference in distributions with continuous random variables, we alternate closed-form updates of the latent indicator variables with SVIGL updates for the continuous flow variables. For the discrete update, we approximate the tedious analytical expectation values over the flow variables with a Monte-Carlo estimator (*c.f*. Eq. 7b). This effectively reduces the optimization w.r.t. the indicator variables to an independent update – thus maintaining the ease of use of SVIGL. Weighting parameters λ_{D} and λ_{S} are determined on a training set with Bayesian optimization [37] using the F1-score as described in [45].

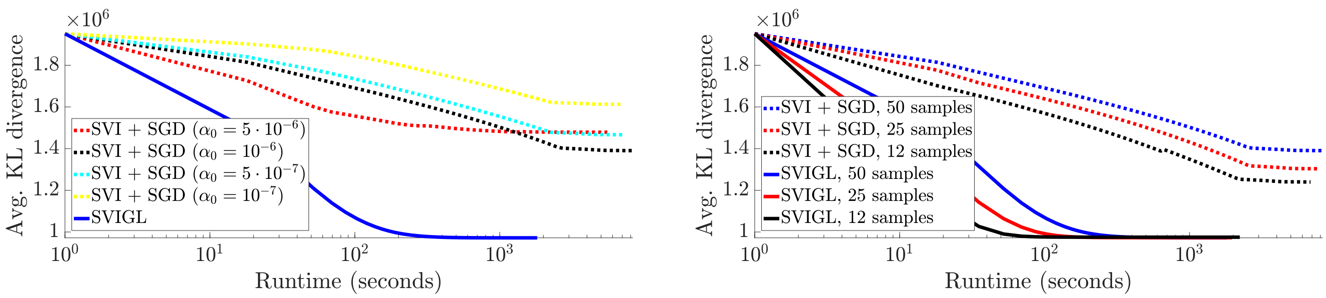
F. Results of Poisson-Gaussian Denoising

Fig. 8 shows some example results of SVIGL applied to Poisson-Gaussian denoising on the BSDS dataset. High uncertainties can be observed especially on object boundaries. Due to the high amount of noise, a strong smoothness term



(a) (b)

Figure 6. Unnormalized KL divergence *vs*. runtime for optical flow with SVIGL and SVI with SGD with different step sizes (a) and different numbers of samples and iterations (b). Values averaged on the validation set.



(a) (b)

Figure 7. Unnormalized KL divergence *vs*. runtime for denoising with SVIGL and SVI with SGD with different step sizes (a) and with different numbers of samples and iterations (b). Values averaged over the BSDS test set.

maximizes the PSNR on the training set. Therefore, the denoised images tend to be rather smooth in general.

G. Results on Sintel Test

As described in Sec. 5.1, we apply SVIGL as well as two MAP baselines on the full-sized Sintel test images in order to evaluate their performance. Figure 9 shows a screenshot of the private Sintel benchmark table with results for both methods. SVIGL outperforms the underlying FlowFields method [1] as well as the L-BFGS baseline and shows an AEPE result on par with the corresponding MAP estimate using GL. Moreover, SVIGL estimates are competitive with the finetuned version of FlowNet2 [49], *i.e*. the state-of-the-art baseline for optical flow prediction with convolutional neural networks.

H. 3D Surface Reconstruction

We now give more details on the application of SVIGL to 3D surface reconstruction. First, we restate the energy of Lipman *et al*. [25], which is given by

$$E(X,P,C) = \sum_{i=1}^{|X|}\sum_{j=1}^{|P|}\|x_i - p_j\| \cdot h\big(\|c_i - p_j\|\big) - \sum_{i=1}^{|X|}\sum_{i'=1}^{|C|}\lambda_i\|x_i - c_{i'}\| \cdot h\big(\|c_i - c_{i'}\|\big). \quad (42)$$

Here, $p_j \in P$ denote the noisy input points, $c_i \in C$ are the current estimates of the smoothed points, and $x_i \in X$ the new estimates of the smoothed points. While the first part of the energy forces the new estimates to be close to the input points, the second term pushes the reconstructed points apart by penalizing points in X that are too close to points in C. The contribution of each term is weighted by the Gaussian kernel $h(\cdot)$.

A closed-form solution to minimizing the above energy is given in [25]. This solution is then used in a fixed point scheme as

$$X_{t+1} = \arg\min_X E(X,P,X_t), \quad (43)$$

where X_0 is initialized as a L_2 projection of the input points.

In a variational inference setting, closed-form updates are no longer possible due to introducing the additional vari-

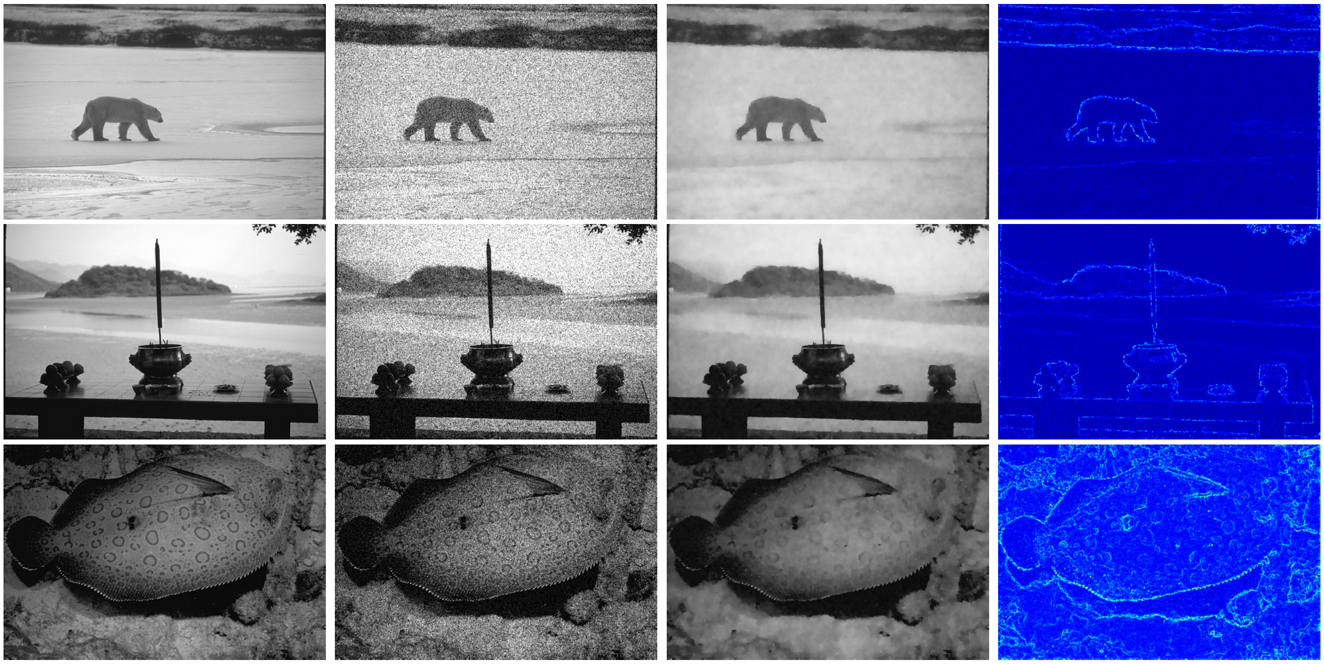


Figure 8. Examples of ground truth (left), noisy images (second column), estimated clean images (third column), and uncertainty estimates (right) from SVIGL on the BSDS test set.

	EPE all	EPE matched	EPE unmatched	d0-10	d10-60	d60-140	s0-10	s10-40	s40+
GroundTruth [1]	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
⋮			⋮						⋮
MAP_GL [25]	5.735	2.524	31.896	4.789	2.126	1.619	1.107	3.659	33.708
FlowNet2-ft-sintel [26]	5.739	2.752	30.108	4.818	2.557	1.735	0.959	3.228	35.538
SVIGL [27]	5.740	2.526	31.924	4.792	2.128	1.619	1.107	3.661	33.740
MAP_LBFGS [28]	5.805	2.626	31.710	4.890	2.233	1.709	1.206	3.803	33.456
FlowFields [29]	5.810	2.621	31.799	4.851	2.232	1.682	1.157	3.739	33.890

Figure 9. Screenshot of the private Sintel benchmark table (final) with results for SVIGL, MAP + GL, MAP + L-BFGS, and the original FlowFields approach [1] (status as of March 2018).

ance variables $\boldsymbol{\sigma}$ of the variational posterior. Hence, we employ SVIGL updates instead. To be able to apply SVIGL, we require a linearization of the energy gradient. The specific form of the energy in Eq. (19) allows for a diagonal linearization:

$$\begin{aligned}\nabla_{x_i} E(X, P, C) = &\sum_{j \in J} (x_i - p_j) \frac{h(\|c_i - p_j\|)}{\|x_i - p_j\|} \\ &- \sum_{i' \in I} (x_i - c_{i'}) \frac{h(\|c_i - c_{i'}\|)}{\|x_i - c_{i'}\|}. \end{aligned} \tag{44}$$

In total, we run 10 iterations of Eq. (43). In each iteration, we compute a single SVIGL update with a sample set size of $|\mathcal{Z}| = 5$.

References

[49] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, pages 1647 – 1655, 2017. 4