Supplementary – Learning Energy Based Inpainting for Optical Flow

Christoph Vogel[1 [0000−0002−5960−1375]], Patrick Knöbelreiter[1 [0000−0002−2371−014X]], and Thomas Pock[1,2 [0000−0001−6120−1058]]

[1] Graz University of Technology, Graz, Austria.
{firstname.lastname}@icg.tugraz.at
[2] Austrian Institute of Technology, Vienna, Austria

Our supplementary material provides the full backward algorithm for inpainting optical flow with Total Generalized Variation [1] of second order, according to Eq. (2) of the main paper. We first repeat the objective in its discretized form:

$$\min_{u_i} \min_{w_i=(w_{i,0},w_{i,1})^\mathsf{T}} \|\sqrt{W}(Du_i - w_i)\|_\delta + \beta(\|Dw_{i,0}\|_\delta + \|Dw_{i,1}\|_\delta) + \|u_i - \hat{u}_i\|_c. \quad (1)$$

Here, the auxiliary variables $w_{i,j} \in \mathbb{R}^{NM}$, $i, j \in \{0, 1\}$ denote the diffusion tensor and the norm $\|\cdot\|_c$ denotes the ℓ_1 norm weighted by the confidence c.

1 Optimization Layer for TGV

For completeness we repeat the forward algorithm first.

$$(u_i^{k+0.5}, w_i^{k+1})^\mathsf{T} := (v_i^k, q_i^k)^\mathsf{T} - B^\mathsf{T} \frac{V_\beta}{\max(1, |\sqrt{V} B(v_i^k, q_i^k)^\mathsf{T}|_2/\delta)} B(v_i^k, q_i^k)^\mathsf{T} \quad (2)$$

$$u_i^{k+1} := \begin{cases} u_i^{k+0.5} - c \text{ if } u_i^{k+0.5} - c > \hat{u}_i \\ u_i^{k+0.5} + c \text{ if } u_i^{k+0.5} + c < \hat{u}_i \\ \hat{u}_i \quad \text{else} \end{cases} \quad (3)$$

$$(v_i^{k+1}, q_i^{k+1})^\mathsf{T} := (u_i^{k+1}, w_i^{k+1})^\mathsf{T} + \frac{t^k - 1}{t^{k+1}} (u_i^{k+1}, w_i^{k+1})^\mathsf{T} - (u_i^k, w_i^k)^\mathsf{T}), \quad (4)$$

Now, analogue to the TV case we can define a single iteration of the backward path for TGV inpainting as follows:

$$\frac{\partial f}{\partial \hat{u}_i} := \frac{\partial f}{\partial \hat{u}_i} + \frac{\partial f}{\partial u_i^{k+1}} \text{ if } c < |\hat{u}_i - u_i^{k+0.5}| \quad (5)$$

$$\frac{\partial f}{\partial c} := \frac{\partial f}{\partial c} + \text{sign}(\hat{u}_i - u_i^{k+0.5}) \frac{\partial f}{\partial u_i^{k+1}} \text{ if } c \geq |\hat{u}_i - u_i^{k+0.5}| \quad (6)$$

$$\frac{\partial f}{\partial u_i^{k+0.5}} := \begin{cases} \frac{\partial f}{\partial u_i^{k+1}} \text{ if } c \geq |\hat{u}_i - u_i^{k+0.5}| \\ 0 \quad \text{else} \end{cases} \tag{7}$$

$$\begin{pmatrix} \frac{\partial f}{\partial v_i^k} \\ \frac{\partial f}{\partial q_i^k} \end{pmatrix} := \left(I - \begin{pmatrix} \frac{\partial}{\partial v_i^k} \\ \frac{\partial}{\partial q_i^k} \end{pmatrix} \left(\frac{V_\beta}{\max\left(1, \left| \sqrt{V} B \begin{pmatrix} v_i^k \\ q_i^k \end{pmatrix} \right|_2 / \delta \right)} B \begin{pmatrix} v_i^k \\ q_i^k \end{pmatrix} \right)^\mathsf{T} \right) B \begin{pmatrix} \frac{\partial f}{\partial u_i^{k+0.5}} \\ \frac{\partial f}{\partial w_i^{k+1}} \end{pmatrix} \tag{8}$$

$$\frac{\partial f}{\partial V_\beta} := \frac{\partial f}{\partial V_\beta} + \frac{\partial}{\partial V_\beta} \left(\frac{V_\beta}{\max\left(1, \left| \sqrt{V} B \begin{pmatrix} v_i^k \\ q_i^k \end{pmatrix} \right|_2 / \delta \right)} B \begin{pmatrix} v_i^k \\ q_i^k \end{pmatrix} \right)^\mathsf{T} B \begin{pmatrix} \frac{\partial f}{\partial u_i^{k+0.5}} \\ \frac{\partial f}{\partial w_i^{k+1}} \end{pmatrix} \tag{9}$$

$$\left(\frac{\partial f}{\partial u_i^k}, \frac{\partial f}{\partial w_i^k} \right)^\mathsf{T} := \left(\frac{\partial f}{\partial u_i^k}, \frac{\partial f}{\partial w_i^k} \right)^\mathsf{T} + \left(1 + \frac{t^{k-1} - 1}{t^k} \right) \left(\frac{\partial f}{\partial v_i^k}, \frac{\partial f}{\partial q_i^k} \right) \tag{10}$$

$$\left(\frac{\partial f}{\partial u_i^{k-1}}, \frac{\partial f}{\partial w_i^{k-1}} \right)^\mathsf{T} := - \frac{t^{k-1} - 1}{t^k} \left(\frac{\partial f}{\partial v_i^k}, \frac{\partial f}{\partial q_i^k} \right)^\mathsf{T}. \tag{11}$$

As before for the TV case, we use the outer products in (8, 9) for a compact notation. Again, our implementation exploits the extreme sparsity of the resulting matrices. To achieve memory and numerical efficiency, the algorithm operates similar to the TV case and we refer to the main paper for a description of the procedure. The only exception is that the gradient w.r.t. β is computed for each pixel and finally has to be summed over all the pixels. Finally, please recall that the algorithm can be easily extended to learn a pixel-wise diffusion tensor for the auxilliary variables w_0, w_1 instead of a single scalar β.

References

1. Bredies, K., Kunisch, K., Pock, T.: Total generalized variation. SIAM J. Imaging Sciences (2010)