Supplementary Material for the SSVM-2013 paper entitled "Convex Generalizations of Total Variation based on the Structure Tensor with Applications to Inverse Problems"

Stamatios Lefkimmiatis[1], Anastasios Roussos[2], Michael Unser[1], and Petros Maragos[3]

[1] Biomedical Imaging Group, EPFL, Lausanne, Switzerland.
[2] School of EECS, Queen Mary University of London, United Kindgdom.
[3] School of ECE, National Technical University of Athens, Greece.

1 Proof of Proposition 1

Let $T(\boldsymbol{x}) = R_\theta \boldsymbol{x}$ denote the rotation of the image coordinates $\boldsymbol{x}$, with R_θ being the rotation matrix. Applying the chain rule to the Jacobian matrix we have that

$$J\{\boldsymbol{u} \circ T\}(\boldsymbol{x}) = J\boldsymbol{u}(T(\boldsymbol{x}))\, R_\theta, \tag{1}$$

where $\circ$ denotes the composition of functions. Now, we write the structure tensor as

$$S_K\{\boldsymbol{u} \circ T\}(\boldsymbol{x}) = R_\theta^T \Big(K * \underbrace{\big((J\boldsymbol{u}(T(\boldsymbol{x})))^T J\boldsymbol{u}(T(\boldsymbol{x})) \big)}_{(h \circ T)(\boldsymbol{x})} \Big) R_\theta. \tag{2}$$

Since the convolution kernel K is rotationally symmetric, it holds that

$$\{K * (h \circ T)\}(\boldsymbol{x}) = \{K * h\}(T(\boldsymbol{x}))\ . \tag{3}$$

Thus, we have

$$S_K\{\boldsymbol{u} \circ T\}(\boldsymbol{x}) = R_\theta^T S_K\{\boldsymbol{u}\}(T(\boldsymbol{x}))\, R_\theta. \tag{4}$$

The structure tensor, evaluated at coordinates $\boldsymbol{x}$, is real and symmetric. Therefore, it admits an eigendecomposition where we can always choose the corresponding eigenvectors to be rotation matrices. This implies that the rotation of the image coordinates leaves untouched the eigenvalues of the structure tensor and affects only its eigenvectors. Since the proposed energy functionals depend only upon the eigevnalues, and since we are integrating over the whole domain, the energy will be preserved despite the transformation of the coordinate system, *i.e.*, $E_p(\boldsymbol{u} \circ T) = E_p(\boldsymbol{u})$.

The contrast covariance property can be verified by noting that a scaling of the image intensities by a positive factor a, corresponds to a scaling of $\sqrt{\lambda_\pm}$ by the same factor. We further use that all ℓ_p norms are one-homogeneous, which completes the proof.

2 Proof of Proposition 2

To find the adjoint of the discrete weighted Jacobian operator, we exploit the relation of the inner products of the spaces $\mathbb{R}^{NM}$ and $\mathcal{X} \triangleq \mathbb{R}^{N\times(LM)\times 2}$

$$\langle \boldsymbol{Y}\,,\,\boldsymbol{J}_K\boldsymbol{u}\rangle_{\mathcal{X}} = \langle \boldsymbol{J}_K^*\boldsymbol{Y}\,,\,\boldsymbol{u}\rangle_2\,. \tag{5}$$

Using the definition of the inner product in $\mathcal{X}$,

$$\langle \boldsymbol{X}\,,\,\boldsymbol{Y}\rangle_{\mathcal{X}} = \sum_{n=1}^{N}\operatorname{tr}\left(\boldsymbol{Y}_n^T\boldsymbol{X}_n\right) \tag{6}$$

we can equivalently write (5) as

$$\sum_{n=1}^{N}\operatorname{tr}\left(\left[\boldsymbol{J}_K\boldsymbol{u}\right]_n^T\boldsymbol{Y}_n\right) = \sum_{m=1}^{M}\sum_{n=1}^{N}\left[\boldsymbol{u}_m\right]_n\left[\boldsymbol{J}_K^*\boldsymbol{Y}\right]_{(n,m)}\,. \tag{7}$$

We then expand the l.h.s of (7), to obtain

$$\begin{aligned}
&\sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{l=1}^{L}\left[P_{\boldsymbol{y}_l}\circ\boldsymbol{D}_h\boldsymbol{u}_m\right]_n\boldsymbol{Y}_n^{((m-1)L+l,1)}+\left[P_{\boldsymbol{y}_l}\circ\boldsymbol{D}_v\boldsymbol{u}_m\right]_n\boldsymbol{Y}_n^{((m-1)L+l,2)}\\
=&\sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{l=1}^{L}\left[\boldsymbol{u}_m\right]_n\left(\left[\boldsymbol{D}_h^*\circ P_{\boldsymbol{y}_l}^*\circ\boldsymbol{Y}^{((m-1)L+l,1)}\right]_n+\left[\boldsymbol{D}_v^*\circ P_{\boldsymbol{y}_l}^*\circ\boldsymbol{Y}^{((m-1)L+l,2)}\right]_n\right)\\
=&\sum_{m=1}^{M}\sum_{n=1}^{N}\left[\boldsymbol{u}_m\right]_n\left(\sum_{l=1}^{L}-\textbf{div}\left[P_{\boldsymbol{y}_l}^*\circ\boldsymbol{Y}^{((m-1)L+l,:)}\right]_n\right),
\end{aligned} \tag{8}$$

where **div** is the discrete divergence[4] and P^* is the adjoint of the shift operator P. Note that $\boldsymbol{Y}_n^{(i,j)}$ with $1\le n\le N$, $1\le i\le LM$, and $1\le j\le 2$, corresponds to a single element of $\boldsymbol{Y}\in\mathcal{X}$, while $\boldsymbol{Y}_n^{(i,:)}$ is a vector whose elements correspond to those of the ith row of the nth matrix component $\boldsymbol{Y}_n\in\mathbb{R}^{(LM)\times 2}$ of $\boldsymbol{Y}$. Now, by comparing the r.h.s of (7) to the r.h.s expansion of (8), it is straightforward to verify that the adjoint of the discrete weighted Jacobian operator is indeed computed according to the formula given in Proposition 2.

3 Proof of Proposition 3

The proof of Proposition 3 is straightforward and it is based on the special structure of the patch-based Jacobian. First, we introduce the variables $\boldsymbol{z}_{m,1}$, $\boldsymbol{z}_{m,2}$ and $\boldsymbol{z}_{m,3}$,

[4] The exact formula for the discrete divergence depends on the discretization scheme one uses for the gradient. For example, in our implementation we used a commonly used discretization that uses forward differences, as in [1]. In this case, the adjoint operator is the discrete divergence that is defined using backward differences (see [1] for details).

defined as

$$\boldsymbol{z}_{m,1}\left[\boldsymbol{x}_n\right] = \left(\boldsymbol{D}_h\left\{\boldsymbol{u}_m\right\}\left[\boldsymbol{x}_n\right]\right)^2 \tag{9a}$$

$$\boldsymbol{z}_{m,2}\left[\boldsymbol{x}_n\right] = \boldsymbol{D}_h\left\{\boldsymbol{u}_m\right\}\left[\boldsymbol{x}_n\right]\cdot\boldsymbol{D}_v\left\{\boldsymbol{u}_m\right\}\left[\boldsymbol{x}_n\right] \tag{9b}$$

$$\boldsymbol{z}_{m,3}\left[\boldsymbol{x}_n\right] = \left(\boldsymbol{D}_v\left\{\boldsymbol{u}_m\right\}\left[\boldsymbol{x}_n\right]\right)^2. \tag{9c}$$

Next, by using the definition of the patch-based Jacobian we have

$$\begin{aligned}
\left[\boldsymbol{J}_K\boldsymbol{u}\right]_n^T\left[\boldsymbol{J}_K\boldsymbol{u}\right]_n &= \begin{bmatrix} \sum_{m=1}^{M}\sum_{l=1}^{L}\boldsymbol{w}\left[\boldsymbol{y}_l\right]^2\boldsymbol{z}_{m,1}\left[\boldsymbol{x}_n-\boldsymbol{y}_l\right] & \sum_{m=1}^{M}\sum_{l=1}^{L}\boldsymbol{w}\left[\boldsymbol{y}_l\right]^2\boldsymbol{z}_{m,2}\left[\boldsymbol{x}_n-\boldsymbol{y}_l\right] \\ \sum_{m=1}^{M}\sum_{l=1}^{L}\boldsymbol{w}\left[\boldsymbol{y}_l\right]^2\boldsymbol{z}_{m,2}\left[\boldsymbol{x}_n-\boldsymbol{y}_l\right] & \sum_{m=1}^{M}\sum_{l=1}^{L}\boldsymbol{w}\left[\boldsymbol{y}_l\right]^2\boldsymbol{z}_{m,3}\left[\boldsymbol{x}_n-\boldsymbol{y}_l\right] \end{bmatrix} \\
&= \begin{bmatrix} \sum_{m=1}^{M}\sum_{l=1}^{L}K\left[\boldsymbol{y}_l\right]\boldsymbol{z}_{m,1}\left[\boldsymbol{x}_n-\boldsymbol{y}_l\right] & \sum_{m=1}^{M}\sum_{l=1}^{L}K\left[\boldsymbol{y}_l\right]\boldsymbol{z}_{m,2}\left[\boldsymbol{x}_n-\boldsymbol{y}_l\right] \\ \sum_{m=1}^{M}\sum_{l=1}^{L}K\left[\boldsymbol{y}_l\right]\boldsymbol{z}_{m,2}\left[\boldsymbol{x}_n-\boldsymbol{y}_l\right] & \sum_{m=1}^{M}\sum_{l=1}^{L}K\left[\boldsymbol{y}_l\right]\boldsymbol{z}_{m,3}\left[\boldsymbol{x}_n-\boldsymbol{y}_l\right] \end{bmatrix} \\
&= K * \begin{bmatrix} \sum_{m=1}^{M}\boldsymbol{z}_{m,1}\left[\boldsymbol{x}_n\right] & \sum_{m=1}^{M}\boldsymbol{z}_{m,2}\left[\boldsymbol{x}_n\right] \\ \sum_{m=1}^{M}\boldsymbol{z}_{m,2}\left[\boldsymbol{x}_n\right] & \sum_{m=1}^{M}\boldsymbol{z}_{m,3}\left[\boldsymbol{x}_n\right] \end{bmatrix} \\
&= K * \begin{bmatrix} \sum_{m=1}^{M}\left(\boldsymbol{D}_h\left\{\boldsymbol{u}_m\right\}\left[\boldsymbol{x}_n\right]\right)^2 & \sum_{m=1}^{M}\boldsymbol{D}_h\left\{\boldsymbol{u}_m\right\}\left[\boldsymbol{x}_n\right]\cdot\boldsymbol{D}_v\left\{\boldsymbol{u}_m\right\}\left[\boldsymbol{x}_n\right] \\ \sum_{m=1}^{M}\boldsymbol{D}_h\left\{\boldsymbol{u}_m\right\}\left[\boldsymbol{x}_n\right]\cdot\boldsymbol{D}_v\left\{\boldsymbol{u}_m\right\}\left[\boldsymbol{x}_n\right] & \sum_{m=1}^{M}\left(\boldsymbol{D}_v\left\{\boldsymbol{u}_m\right\}\left[\boldsymbol{x}_n\right]\right)^2 \end{bmatrix} \\
&= \left[\boldsymbol{S}_K\boldsymbol{u}\right]_n.
\end{aligned} \tag{10}$$

4 MFISTA Framework for General Linear Inverse Problems

A detailed description of the overall algorithm for solving general linear inverse problems of the form

$$\arg\min_{\boldsymbol{u}\in\mathbb{R}^{NM}} \frac{1}{2}\left\|\boldsymbol{A}\boldsymbol{u}-\boldsymbol{z}\right\|_2^2 + \underbrace{\tau E_p\left(\boldsymbol{u}\right)+\iota_{\mathcal{C}}\left(\boldsymbol{u}\right)}_{\psi(\boldsymbol{u})}, \forall p \geq 1, \tag{11}$$

is provided in Algorithm 2. We note that the proposed minimization strategy depends on the evaluation of the proximal map of the regularizer $\psi\left(\boldsymbol{u}\right)$. This evaluation takes place using the proposed method described in Algorithm 1.

Algorithm 2 : Numerical algorithm for solving general linear inverse problems.

Input: $\boldsymbol{y}$, $\boldsymbol{A}$, $\tau > 0$, $p \geq 1$, $\alpha > \left\|\boldsymbol{A}^T\boldsymbol{A}\right\|$, $\Pi_{\mathcal{C}}$.

Initialization: $\boldsymbol{v}_1 = \boldsymbol{u}_0$, $t_1 = 1$, $c_1 = \psi\left(\boldsymbol{u}_0\right)$.

while *stopping criterion is not satisfied* **do**

- $\boldsymbol{z}_n \leftarrow \mathtt{prox}_{\frac{\tau}{\alpha}\psi}\left(\boldsymbol{v}_n + \alpha^{-1}\boldsymbol{A}^T\left(\boldsymbol{y}-\boldsymbol{A}\boldsymbol{v}_n\right)\right)$;
- $t_{n+1} \leftarrow \frac{1+\sqrt{1+4t_n^2}}{2}$;
- $c_{n+1} = \psi\left(\boldsymbol{z}_n\right)$;
- **if** $c_{n+1} > c_n$ **then**
  - $c_{n+1} = c_n$;
  - $\boldsymbol{u}_{n+1} \leftarrow \boldsymbol{u}_n$;
  - $\boldsymbol{v}_{n+1} \leftarrow \boldsymbol{u}_n + \frac{t_n}{t_{n+1}}\left(\boldsymbol{z}_n - \boldsymbol{u}_n\right)$;
- **else**
  - $\boldsymbol{u}_{n+1} \leftarrow \boldsymbol{z}_n$;
  - $\boldsymbol{v}_{n+1} \leftarrow \boldsymbol{z}_n + \left(\frac{t_n-1}{t_{n+1}}\right)\left(\boldsymbol{z}_n - \boldsymbol{u}_n\right)$;
- **end**
- $n \leftarrow n+1$;

end

return $\boldsymbol{u}_n$;

References

1. Chambolle, A.: An algorithm for total variation minimization and applications. J. Math. Imaging and Vision **20** (2004) 89–97