ADMM for snapshot foreground-background separation

1 Problem formulation

Our goal is to solve

$$\begin{array}{ll} \underset{\boldsymbol{S_1},\ldots,\boldsymbol{S_F}}{\text{minimize}} & \|\boldsymbol{S_1}\|_1 + \cdots + \|\boldsymbol{S_F}\|_1 + r\Big(\boldsymbol{B}, \boldsymbol{S_1}, \ldots, \boldsymbol{S_F}\Big) \\ \text{subject to} & \boldsymbol{Y_S} = \mathcal{H}(\boldsymbol{S_1}, \ldots, \boldsymbol{S_F})\,, \end{array} \tag{1}$$

where r is a prior that depends on the known background $\boldsymbol{B}$ and sparse foreground frames $\boldsymbol{S_1}$, $\ldots$, $\boldsymbol{S_F}$. Once we find a solution $(\widehat{\boldsymbol{S}}_1, \ldots, \widehat{\boldsymbol{S}}_F)$ of (1), the final scene can be recovered as

$$\Big(\widehat{\boldsymbol{X}}_{\boldsymbol{1}}, \ldots, \widehat{\boldsymbol{X}}_{\boldsymbol{F}}\Big) = \mathbf{1}_{1\times F} \otimes \boldsymbol{B} + \Big(\widehat{\boldsymbol{S}}_{\boldsymbol{1}}, \ldots, \widehat{\boldsymbol{S}}_{\boldsymbol{F}}\Big) = \Big(\boldsymbol{B} + \widehat{\boldsymbol{S}}_{\boldsymbol{1}}, \ldots, \boldsymbol{B} + \widehat{\boldsymbol{S}}_{\boldsymbol{F}}\Big)\,.$$

2 ADMM

The ADMM algorithm [1] solves problems with the format

$$\begin{array}{ll} \underset{\boldsymbol{y},\boldsymbol{z}}{\text{minimize}} & f(\boldsymbol{y}) + g(\boldsymbol{z}) \\ \text{subject to} & \boldsymbol{F}\boldsymbol{y} + \boldsymbol{G}\boldsymbol{z} = \mathbf{0}\,, \end{array} \tag{2}$$

where f and g are closed, proper, and convex functions, and $\boldsymbol{F}$ and $\boldsymbol{G}$ are given matrices. Associating a dual variable $\boldsymbol{\lambda}$ to the constraints of (2), ADMM iterates on k

$$\boldsymbol{y}^{k+1} = \underset{\boldsymbol{y}}{\text{argmin}}\; f(\boldsymbol{y}) + \frac{\rho}{2}\Big\|\boldsymbol{F}\boldsymbol{y} + \boldsymbol{G}\boldsymbol{z}^k + \boldsymbol{\lambda}^k\Big\|_2^2 \tag{3a}$$

$$\boldsymbol{z}^{k+1} = \underset{\boldsymbol{z}}{\text{argmin}}\; g(\boldsymbol{z}) + \frac{\rho}{2}\Big\|\boldsymbol{F}\boldsymbol{y}^{k+1} + \boldsymbol{G}\boldsymbol{z} + \boldsymbol{\lambda}^k\Big\|_2^2 \tag{3b}$$

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \boldsymbol{F}\boldsymbol{y}^{k+1} + \boldsymbol{G}\boldsymbol{z}^{k+1}\,, \tag{3c}$$

where $\rho > 0$ is the augmented Lagrangian parameter. The augmented Lagrangian is

$$L_\rho(\boldsymbol{y}, \boldsymbol{z}; \boldsymbol{\lambda}) = f(\boldsymbol{y}) + g(\boldsymbol{z}) + \boldsymbol{\lambda}^\top(\boldsymbol{F}\boldsymbol{y} + \boldsymbol{G}\boldsymbol{z}) + \frac{\rho}{2}\|\boldsymbol{F}\boldsymbol{y} + \boldsymbol{G}\boldsymbol{z}\|_2^2\,. \tag{4}$$

Note that ADMM (3) consists of iteratively minimizing L_ρ w.r.t. $\boldsymbol{y}$, then w.r.t. $\boldsymbol{z}$ (with $\boldsymbol{y}$ fixed at the new value), and updating the dual variable $\boldsymbol{\lambda}$ (gradient ascent on the dual function).

3 Application of ADMM

To apply ADMM to (1), we need to rewrite it in the same format as (2). To do so, we introduce two copies of the variables $\boldsymbol{S_f}$: $\boldsymbol{U_f}$ and $\boldsymbol{V_f}$, $f = 1, \dots, F$. Problem (1) is thus equivalent to

$$\begin{array}{ll} \underset{\{\boldsymbol{S_f}, \boldsymbol{U_f}, \boldsymbol{V_f}\}_{f=1}^F}{\text{minimize}} & \sum_{f=1}^F \|\boldsymbol{U_f}\|_1 + r\Big(\boldsymbol{B}, \boldsymbol{V_1}, \dots, \boldsymbol{V_F}\Big) \\ \text{subject to} & \boldsymbol{Y_S} = \mathcal{H}(\boldsymbol{S_1}, \dots, \boldsymbol{S_F}) \\ & \boldsymbol{S_f} = \boldsymbol{U_f}\,, \qquad f = 1, \dots, F \\ & \boldsymbol{S_f} = \boldsymbol{V_f}\,, \qquad f = 1, \dots, F\,. \end{array} \tag{5}$$

We establish the following correspondence between (1) and (5):

$$\begin{aligned} \boldsymbol{y} &\leftarrow \{\boldsymbol{S_f}\}_f \\ \boldsymbol{z} &\leftarrow \Big(\{\boldsymbol{U_f}\}_f, \{\boldsymbol{V_f}\}_f\Big) \\ f(\boldsymbol{y}) &= \mathrm{i}_{\mathcal{H}}(\boldsymbol{S_1}, \dots, \boldsymbol{S_F}) \\ g(\boldsymbol{z}) &= \sum_{f=1}^F \|\boldsymbol{U_f}\|_1 + r\Big(\boldsymbol{B}, \boldsymbol{V_1}, \dots, \boldsymbol{V_F}\Big)\,, \end{aligned}$$

where $\mathrm{i}_{\mathcal{H}}$ is the indicator function of the first constraint of (5), i.e.,

$$\mathrm{i}_{\mathcal{H}}(\boldsymbol{S_1}, \dots, \boldsymbol{S_F}) = \begin{cases} 0 & \text{if } \boldsymbol{Y_S} = \mathcal{H}(\boldsymbol{S_1}, \dots, \boldsymbol{S_F}) \\ +\infty & \text{if } \boldsymbol{Y_S} \neq \mathcal{H}(\boldsymbol{S_1}, \dots, \boldsymbol{S_F}). \end{cases}$$

Writing the correspondences for the matrices $\boldsymbol{F}$ and $\boldsymbol{G}$ is cumbersome, since the variables are matrices in (1) and vectors in (2). The easiest way to apply ADMM to (5) is to first write the corresponding augmented Lagrangian (4), using $\langle \boldsymbol{A}, \boldsymbol{B} \rangle = \mathrm{tr}(\boldsymbol{A}^\top \boldsymbol{B})$ as the matrix inner product between two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, and the corresponding induced Frobenius norm $\|\boldsymbol{A}\|_F := \sqrt{\langle \boldsymbol{A}, \boldsymbol{A} \rangle} = \sqrt{\mathrm{tr}(\boldsymbol{A}^\top \boldsymbol{A})}$. Using the shorthand notation $\overline{\boldsymbol{S}} := \{\boldsymbol{S_f}\}_{f=1}^F$, $\overline{\boldsymbol{U}} := \{\boldsymbol{U_f}\}_{f=1}^F$, and $\overline{\boldsymbol{V}} := \{\boldsymbol{V_f}\}_{f=1}^F$ and associating the matrices $\overline{\boldsymbol{\Lambda}} = \{\boldsymbol{\Lambda_f}\}_f^F$ and $\overline{\boldsymbol{\Gamma}} = \{\boldsymbol{\Gamma_f}\}_f^F$ to the second and third constraints of (5), this yields

$$\begin{aligned} L_\rho\Big(\overline{\boldsymbol{S}}, \overline{\boldsymbol{U}}, \overline{\boldsymbol{V}}\,;\, \overline{\boldsymbol{\Lambda}}, \overline{\boldsymbol{\Gamma}}\Big) = r\Big(\boldsymbol{B}, \overline{\boldsymbol{V}}\Big) + \mathrm{i}_{\mathcal{H}}\Big(\overline{\boldsymbol{S}}\Big) + \sum_{f=1}^F \Bigg[\|\boldsymbol{U_f}\|_1 + \mathrm{tr}\Big(\boldsymbol{\Lambda_f}^\top(\boldsymbol{S_f} - \boldsymbol{U_f})\Big) + \mathrm{tr}\Big(\boldsymbol{\Gamma_f}^\top(\boldsymbol{S_f} - \boldsymbol{V_f})\Big) \\ + \frac{\rho}{2}\Big\|\boldsymbol{S_f} - \boldsymbol{U_f}\Big\|_F^2 + \frac{\rho}{2}\Big\|\boldsymbol{S_f} - \boldsymbol{V_f}\Big\|_F^2 \Bigg]. \end{aligned} \tag{6}$$

As observed for (3)-(4), ADMM iteratively minimizes L_ρ w.r.t. the first set of variables $\overline{\boldsymbol{S}}$; then, with $\overline{\boldsymbol{S}}$ fixed at the new value $\overline{\boldsymbol{S}}^{k+1}$, minimizes L_ρ w.r.t. the second set of variables $(\overline{\boldsymbol{U}}, \overline{\boldsymbol{V}})$ [which will decompose into two sets of problems that can be solved in parallel]; then, it updates the Lagrange multipliers $\overline{\boldsymbol{\Lambda}}$ and $\overline{\boldsymbol{\Gamma}}$.

More concretely, starting with arbitrary $\Big(\overline{\boldsymbol{U}}^0, \overline{\boldsymbol{V}}^0\Big)$ and $\Big(\overline{\boldsymbol{\Lambda}}^0, \overline{\boldsymbol{\Gamma}}^0\Big)$, ADMM iterates for $k = 0, 1, \dots,$

$$\overline{\boldsymbol{S}}^{k+1} = \underset{\overline{\boldsymbol{S}}}{\arg\min}\ \ \mathrm{i}_{\mathcal{H}}\Big(\overline{\boldsymbol{S}}\Big) + \sum_{f=1}^F \Bigg[\mathrm{tr}\Big((\boldsymbol{\Lambda_f^k} + \boldsymbol{\Gamma_f^k})^\top \boldsymbol{S_f}\Big) + \frac{\rho}{2}\Big\|\boldsymbol{S_f} - \boldsymbol{U_f^k}\Big\|_F^2 + \frac{\rho}{2}\Big\|\boldsymbol{S_f} - \boldsymbol{V_f^k}\Big\|_F^2 \Bigg] \tag{7a}$$

$$\left(\overline{\boldsymbol{U}}^{k+1}, \overline{\boldsymbol{V}}^{k+1}\right) = \underset{\overline{\boldsymbol{U}}, \overline{\boldsymbol{V}}}{\arg\min} \;\; \sum_{f=1}^{F} \left[\|\boldsymbol{U_f}\|_1 - \mathrm{tr}\left(\boldsymbol{\Lambda_f^k}^\top \boldsymbol{U_f}\right) + \frac{\rho}{2}\left\|\boldsymbol{U_f} - \boldsymbol{S_f^{k+1}}\right\|_F^2 \right] + r\left(\boldsymbol{B}, \overline{\boldsymbol{V}}\right)$$
$$+ \sum_{f=1}^{F} \left[\frac{\rho}{2}\left\|\boldsymbol{V_f} - \boldsymbol{S_f^{k+1}}\right\|_F^2 - \mathrm{tr}\left(\boldsymbol{\Gamma_f^k}^\top \boldsymbol{V_f}\right) \right] \tag{7b}$$
$$\left(\overline{\boldsymbol{\Lambda}}^{k+1}, \overline{\boldsymbol{\Gamma}}^{k+1}\right) = \left(\overline{\boldsymbol{\Lambda}}^{k}, \overline{\boldsymbol{\Gamma}}^{k}\right) + \rho\left(\overline{\boldsymbol{S}}^{k+1} - \overline{\boldsymbol{U}}^{k+1}, \overline{\boldsymbol{S}}^{k+1} - \overline{\boldsymbol{V}}^{k+1}\right). \tag{7c}$$

We now elaborate on how to solve each problem in (7).

Solving (7a). Problem (7a) is a projection onto the set of feasible solutions in (1), i.e., onto the set defined by $\boldsymbol{Y_S} = \mathcal{H}(\boldsymbol{S_1}, \ldots, \boldsymbol{S_F})$. Note that the second term of (7a) can be rewritten as

$$\rho \sum_{f=1}^{F} \left\| \boldsymbol{S_f} - \left[\frac{1}{2}\boldsymbol{U_f^k} + \frac{1}{2}\boldsymbol{V_f^k} - \frac{1}{2\rho}(\boldsymbol{\Lambda_f^k} + \boldsymbol{\Gamma_f^k})\right] \right\|_F^2 + \cdots ,$$

where we omitted terms that do not depend on $\boldsymbol{S_f}$. Let us define

$$\boldsymbol{P_f^k} := \frac{1}{2}\boldsymbol{U_f^k} + \frac{1}{2}\boldsymbol{V_f^k} - \frac{1}{2\rho}(\boldsymbol{\Lambda_f^k} + \boldsymbol{\Gamma_f^k}), \qquad f = 1, \ldots, F, \tag{8}$$

and just like for the other variables, let $\overline{\boldsymbol{P}}^k$ denote the set of all the $\boldsymbol{P_f^k}$s: $\overline{\boldsymbol{P}}^k := \{\boldsymbol{P_f^k}\}_{f=1}^F$. If we vectorize $\boldsymbol{s} := \mathrm{vec}(\overline{\boldsymbol{S}})$, $\boldsymbol{p} := \mathrm{vec}(\overline{\boldsymbol{P}})$ and $\boldsymbol{y_S} := \mathrm{vec}(\boldsymbol{Y_S})$, problem (7a) can be written as

$$\begin{aligned} \boldsymbol{s}^{k+1} = \;\; & \underset{\boldsymbol{s}}{\arg\min} && \tfrac{1}{2}\|\boldsymbol{s} - \boldsymbol{p}^k\|_2^2 \\ & \text{s.t.} && \boldsymbol{y_S} = \boldsymbol{H}\boldsymbol{s}, \end{aligned} \tag{9}$$

where $\boldsymbol{H}$ implements the operator $\mathcal{H}$ when its input and output are vectorized. Assuming that $\boldsymbol{H}$ has full row-rank, i.e., $\boldsymbol{H}\boldsymbol{H}^\top$ is invertible, (9) has the closed-form solution

$$\boldsymbol{s}^{k+1} = \boldsymbol{p}^k - \boldsymbol{H}^\top(\boldsymbol{H}\boldsymbol{H}^\top)^{-1}(\boldsymbol{H}\boldsymbol{p}^k - \boldsymbol{y_s}). \tag{10}$$

When the snapshot measurement operator $\mathcal{H}$ implements no shifts, i.e., it just sums all the frames (possibly with a mask), the matrix $\boldsymbol{H}\boldsymbol{H}^\top$ is diagonal (see [2, eq.19]). In that case, (10) can be implemented efficiently.

Question: is it the same for the one-pixel shift operator?

Solving (7b). The problems in $\overline{\boldsymbol{U}}$ and $\overline{\boldsymbol{V}}$ in (7b) are decoupled: the first term in its objective depends only on $\overline{\boldsymbol{U}}$, and the second one depends only on $\overline{\boldsymbol{V}}$. Thus, (7b) decouples into two sets of problems that can be solved in parallel.

Problem in $\overline{\boldsymbol{U}}$. To simplify the problem, we first notice that for each $f = 1, \ldots, F$,

$$\begin{aligned} & \|\boldsymbol{U_f}\|_1 - \mathrm{tr}\left(\boldsymbol{\Lambda_f^k}^\top \boldsymbol{U_f}\right) + \frac{\rho}{2}\left\|\boldsymbol{U_f} - \boldsymbol{S_f^{k+1}}\right\|_F^2 \\ = \; & \|\boldsymbol{U_f}\|_1 + \frac{\rho}{2}\left[\left\|\boldsymbol{U_f}\right\|_F^2 - 2\mathrm{tr}\left(\boldsymbol{S_f^{k+1}} - \frac{1}{\rho}\boldsymbol{\Lambda_f^k}\right)^\top \boldsymbol{U_f}\right] + \cdots \\ = \; & \|\boldsymbol{U_f}\|_1 + \frac{\rho}{2}\left\|\boldsymbol{U_f} - \left(\boldsymbol{S_f^{k+1}} - \frac{1}{\rho}\boldsymbol{\Lambda_f^k}\right)\right\|_F^2 + \cdots , \end{aligned}$$

where we omitted terms not depending on $\boldsymbol{U_f}$. Let

$$\boldsymbol{Q}_{\boldsymbol{f}}^{k} := \boldsymbol{S}_{\boldsymbol{f}}^{k+1} - \frac{1}{\rho}\boldsymbol{\Lambda}_{\boldsymbol{f}}^{k}\,. \tag{11}$$

The problem in $\overline{\boldsymbol{U}}$ in (7b) is thus

$$\overline{\boldsymbol{U}}^{k+1} = \underset{\overline{\boldsymbol{U}}}{\arg\min}\ \sum_{f=1}^{F}\|\boldsymbol{U_f}\|_1 + \frac{\rho}{2}\left\|\boldsymbol{U_f} - \boldsymbol{Q}_{\boldsymbol{f}}^{k}\right\|_F^2,$$

which decomposes independent problems not only across frames, but also across pixels. Denoting by u (resp. q) a generic entry from $\overline{\boldsymbol{U}}$ (resp. $\overline{\boldsymbol{Q}}^k := \{\boldsymbol{Q}_{\boldsymbol{f}}^k\}_{f=1}^F$), each problem has the format

$$\begin{aligned} u^{k+1} &= \underset{u}{\arg\min}\ |u| + \frac{\rho}{2}(u-q)^2 \\ &= \begin{cases} q+\frac{1}{\rho} & ,\ q < -\frac{1}{\rho} \\ 0 & ,\ |q| \le \frac{1}{\rho} \\ q-\frac{1}{\rho} & ,\ q > \frac{1}{\rho}\,, \end{cases} \end{aligned} \tag{12}$$

that is, a soft-thresholding solution.

Problem in $\overline{\boldsymbol{V}}$. Given the similarity between the quadratic+linear terms in $\boldsymbol{U_f}$ and $\boldsymbol{V_f}$ in (7b), simplify the problem in a similar way, i.e., we first write

$$-\mathrm{tr}\Big({\boldsymbol{\Gamma}_{\boldsymbol{f}}^{\boldsymbol{k}}}^{\top}\boldsymbol{V_f}\Big) + \frac{\rho}{2}\left\|\boldsymbol{V_f} - \boldsymbol{S}_{\boldsymbol{f}}^{k+1}\right\|_F^2 = \frac{\rho}{2}\left\|\boldsymbol{V_f} - \Big(\boldsymbol{S}_{\boldsymbol{f}}^{k+1} - \frac{1}{\rho}\boldsymbol{\Gamma}_{\boldsymbol{f}}^{k}\Big)\right\|_F^2 + \cdots,$$

where, again, we omitted terms independent from $\boldsymbol{V_f}$. Defining

$$\boldsymbol{R}_{\boldsymbol{f}}^{k} := \boldsymbol{S}_{\boldsymbol{f}}^{k+1} - \frac{1}{\rho}\boldsymbol{\Gamma}_{\boldsymbol{f}}^{k}\,, \tag{13}$$

the problem in $\overline{\boldsymbol{V}}$ in (7b) is

$$\overline{\boldsymbol{V}}^{k+1} = \underset{\overline{\boldsymbol{V}}}{\arg\min}\ r\Big(\boldsymbol{B}, \overline{\boldsymbol{V}}\Big) + \frac{\rho}{2}\sum_{f=1}^{F}\left\|\boldsymbol{V_f} - \boldsymbol{R_f}\right\|_F^2. \tag{14}$$

When r is the nuclear norm of the matrix whose columns are patches extracted from the entire snapshot sequence, this problem corresponds to problems (7)/(14)/(24) in [2] and thus can be solved using similar techniques.

4 Resulting algorithm

Notice that the dual variables $\boldsymbol{\Lambda}_{\boldsymbol{f}}^{k}$ and $\boldsymbol{\Gamma}_{\boldsymbol{f}}^{k}$ in (8), (11), and (13) always appear multiplied by $1/\rho$. If we multiply both sides of (7c) by $1/\rho$, all their appearances in the algorithm is multiplied by that factor. Hence, we redefine them as

$$\begin{aligned} \boldsymbol{\Lambda}_{\boldsymbol{f}}^{k} &\leftarrow \frac{1}{\rho}\boldsymbol{\Lambda}_{\boldsymbol{f}}^{k} \\ \boldsymbol{\Gamma}_{\boldsymbol{f}}^{k} &\leftarrow \frac{1}{\rho}\boldsymbol{\Gamma}_{\boldsymbol{f}}^{k}\,. \end{aligned}$$

The resulting algorithm is described in Algorithm 1. See an example of the implementation of ADMM to solve a simpler problem in Github.

Algorithm 1 ADMM applied to (5)

Input: measurements $\boldsymbol{Y_S} \in \mathbb{R}^{M\times(N+F-1)}$, initial $\rho > 0$, dimensions (M, N, F), matrix $\boldsymbol{H}$ (or function handles for matrix-vector products), background $\boldsymbol{B} \in \mathbb{R}^{M\times N}$, maximum # of iterations K, and stopping tolerance $\epsilon > 0$

Initialization: Set $\boldsymbol{U}_{\boldsymbol{f}}^0, \boldsymbol{V}_{\boldsymbol{f}}^0, \boldsymbol{\Lambda}_{\boldsymbol{f}}^0, \boldsymbol{\Gamma}_{\boldsymbol{f}}^0 \in \mathbb{R}^{M\times N}$, for $f = 1, \ldots, F$, arbitrarily

1: **for** $k = 0, 1, \ldots, K$ **do**

2: Set $\boldsymbol{P}_{\boldsymbol{f}}^k = \frac{1}{2}\boldsymbol{U}_{\boldsymbol{f}}^k + \frac{1}{2}\boldsymbol{V}_{\boldsymbol{f}}^k - \frac{1}{2}(\boldsymbol{\Lambda}_{\boldsymbol{f}}^k + \boldsymbol{\Gamma}_{\boldsymbol{f}}^k)$, for $f = 1, \ldots, F$

3: Find

$$\boldsymbol{s}^{k+1} = \boldsymbol{p}^k - \boldsymbol{H}^\top(\boldsymbol{H}\boldsymbol{H}^\top)^{-1}(\boldsymbol{H}\boldsymbol{p}^k - \boldsymbol{y_s}),$$

where $\boldsymbol{s}^{k+1}$, $\boldsymbol{p}^k$, and $\boldsymbol{y_s}$ are vectorizations of $\overline{\boldsymbol{S}}^{k+1}$, $\overline{\boldsymbol{P}}^k$, and $\boldsymbol{Y_S}$

4: Set, for all $f = 1, \ldots, F$,

$$\boldsymbol{Q}_{\boldsymbol{f}}^k = \boldsymbol{S}_{\boldsymbol{f}}^{k+1} - \boldsymbol{\Lambda}_{\boldsymbol{f}}^k$$
$$\boldsymbol{R}_{\boldsymbol{f}}^k = \boldsymbol{S}_{\boldsymbol{f}}^{k+1} - \boldsymbol{\Gamma}_{\boldsymbol{f}}^k$$

5: Each entry of $\boldsymbol{U}_{\boldsymbol{f}}^{k+1}$ is computed via soft-thresholding (12), using $\boldsymbol{Q}_{\boldsymbol{f}}^k$ and ρ, $f = 1, \ldots, F$

6: The variables $\boldsymbol{V_f}$ are updated by solving (14), using $\boldsymbol{B}$, $\boldsymbol{R_f}$ and ρ

7: Update dual variables, for all $f = 1, \ldots, F$,

$$\boldsymbol{\Lambda}_{\boldsymbol{f}}^{k+1} = \boldsymbol{\Lambda}_{\boldsymbol{f}}^k + \boldsymbol{S}_{\boldsymbol{f}}^{k+1} - \boldsymbol{U}_{\boldsymbol{f}}^{k+1}$$
$$\boldsymbol{\Gamma}_{\boldsymbol{f}}^{k+1} = \boldsymbol{\Gamma}_{\boldsymbol{f}}^k + \boldsymbol{S}_{\boldsymbol{f}}^{k+1} - \boldsymbol{V}_{\boldsymbol{f}}^{k+1}$$

8: Compute primal and dual residuals, and update ρ [1, §3.4.1]

9: **if** dual + primal residuals $< \epsilon$ **then**

10: Break

11: **end if**

12: **end for**

References

[1] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating method of multipliers,” *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[2] Y. Liu, X. Yuan, J. Suo, D. J. Brady, and Q. Dai, “Rank minimization for snapshot compressive imaging,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 41, no. 12, pp. 2990–3006, 2019. DOI: 10.1109/TPAMI.2018.2873587.