Robust image reconstruction from multi-view measurements[*]

Gilles Puy [†] and Pierre Vandergheynst [†]

Abstract. We propose a novel method to accurately reconstruct a set of images representing a single scene from few linear multi-view measurements. Each observed image is modeled as the sum of a background image and a foreground one. The background image is common to all observed images but undergoes geometric transformations, as the scene is observed from different viewpoints. In this paper, we assume that these geometric transformations are represented by a few parameters, e.g., translations, rotations, affine transformations, etc.. The foreground images differ from one observed image to another, and are used to model possible occlusions of the scene. The proposed reconstruction algorithm estimates jointly the images and the transformation parameters from the available multi-view measurements. The ideal solution of this multi-view imaging problem minimizes a non-convex functional, and the reconstruction technique is an alternating descent method built to minimize this functional. The convergence of the proposed algorithm is studied, and conditions under which the sequence of estimated images and parameters converges to a critical point of the non-convex functional are provided. Finally, the efficiency of the algorithm is demonstrated using numerical simulations for applications such as compressed sensing or super-resolution.

Key words. compressed sensing, inverse problem, non-convex optimization, super-resolution, robust-image alignment.

AMS subject classifications. 49N45, 90C26, 90C90.

1. Introduction. Multi-view imaging has become more and more popular in the signal processing community during the past years with, for example, the design of camera arrays [1] and the rise of new applications such as 3D reconstruction of a scene [21].

In most multi-view imaging applications, the pose of the cameras, or of the sensing systems, is estimated by pre-calibration or by using side-information. In this paper, we study the problem of reconstructing a set of images, representing a single scene, from few measurements made at different viewpoints in absence of (or with inaccurate) prior pose estimation. This situation may occur in different scenarios.

For example, let us imagine that we have a camera that can be sent up to take pictures of the ground and that we are interested in obtaining higher resolution images from these acquisitions. Without more knowledge on the acquisition strategy, we can treat each image independently using a single-frame super-resolution technique. However, if we know that the duration between the capture of two images is not too long, it is highly probable that parts of the scene visible in one frame remain visible in the subsequent ones. Consequently, we can surely benefit from the high inter-correlation between observations to obtain better super-resolved images by jointly reconstructing several frames together, as in, e.g., [18, 44]. The inter-correlation may be modeled using geometric transformations, which depend on the camera pose, that register the images with respect to each other. In the absence of any side information, these transformations have to be estimated along with the high resolution images

---

[†]Institute of Electrical Engineering, Ecole Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland. E-Mail: gilles.puy@epfl.ch and pierre.vandergheynst@epfl.ch.

during the reconstruction process.

As another example, let us consider the problem of designing a similar camera as above but with strong power consumption constraints. To design such an energy-efficient system, compressed sensing (CS) is a powerful tool [12, 23]. Indeed, this theory states that sparse signals can be sampled with just a few linear and non-adaptive measurements. In scenarios where signals related to common phenomena are acquired, it is actually possible to reduce even further the number of measurements by jointly reconstructing the signals with a method exploiting the inter-correlation between them [6, 16]. A technique, as in [24, 45], able to estimate both the geometric transformations between the observed images and the images themselves from the compressed measurements can thus help us to meet our power consumption constraints.

In the present work, we propose a novel method that estimates jointly a set of correlated images and the geometric transformations that align these images on each other. This technique also robustly handles the appearance of new objects in the scene and can be applied in different settings, such as the ones described above.

1.1. Problem formulation. Let $\boldsymbol{u} = (u_1, u_2) \in \mathbb{R}^2$ represent the Cartesian coordinates of a point on the Euclidean plane, and $G = \{\boldsymbol{u}_k\}_{1 \leqslant k \leqslant n}$ be a square uniform grid of n points. We model continuous images as functions $x \colon \mathbb{R}^2 \to \mathbb{R}$ living in a space $\{x(\boldsymbol{u}) = \sum_{i=1}^n x_i \; \varphi(\boldsymbol{u} - \boldsymbol{u}_i), \; x_i \in \mathbb{R}, \; \boldsymbol{u}_i \in G, \; i = 1, \ldots, n\}$, where $\varphi \colon \mathbb{R}^2 \to \mathbb{R}$ is a generating function [43]. For simplicity, we suppose that $\varphi(\boldsymbol{u}_k - \boldsymbol{u}_i) = \delta(\boldsymbol{u}_k - \boldsymbol{u}_i)$ for all $\boldsymbol{u}_k, \boldsymbol{u}_i \in G$, so that the discrete image $\boldsymbol{x} \in \mathbb{R}^n$ obtained by sampling x on the grid G has samples $x_1, \ldots, x_n$.

In our setting, several observers provide different observations $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_l \in \mathbb{R}^m$, with $m \leqslant n$, of a scene from different viewpoints. As a first approximation, one can consider that these observations describe a single image x_0, referred as the *background* image hereafter. This image is "viewed" from different perspectives and thus undergoes geometric transformations. We consider here that these transformations are unknown and need to be estimated. However, we restrict ourselves to global transformations represented by few parameters, such as translations, rotations, or affine transformations. For each observer j, with $j = 1, \ldots, l$, these transformations are modeled using a function $\tau_{\boldsymbol{\theta}_j} \colon \mathbb{R}^2 \to \mathbb{R}^2$, depending on q parameters $\boldsymbol{\theta}_j \in \mathbb{R}^q$, that maps the coordinates $\boldsymbol{u}$ into new coordinates $\tau_{\boldsymbol{\theta}_j}(\boldsymbol{u})$. The background image transformed by $\tau_{\boldsymbol{\theta}_j}$ is thus $x_0 \circ \tau_{\boldsymbol{\theta}_j}$.

To complete the model, we also take into account possible occlusions of the scene. These occlusions may obviously be different from one observer to another. We model them here using l *foreground* images $x_1, \ldots, x_l$ and write the jth observed image as $x_0 \circ \tau_{\boldsymbol{\theta}_j} + x_j$.

Finally, we assume that the observations $\boldsymbol{y}_j$ are obtained by linear projection of $x_0 \circ \tau_{\boldsymbol{\theta}_j} + x_j$ onto m known functions $a_1^j, \ldots, a_m^j \colon \mathbb{R}^2 \to \mathbb{R}$. These functions model the acquisition system. They can represent a blurring operator for a deconvolution problem, or random waveforms in a compressed sensing setting. The ith entry of $\boldsymbol{y}_j$ satisfies

$$
\begin{aligned}
y_{ij} &= \int_{\mathbb{R}^2} \left(x_0 \circ \tau_{\boldsymbol{\theta}_j}(\boldsymbol{u}) + x_j(\boldsymbol{u}) \right) a_i^j(\boldsymbol{u}) \; \mathrm{d}\boldsymbol{u} \\
&= \int_{\mathbb{R}^2} \sum_{k=1}^n (x_k^0 \; \varphi(\tau_{\boldsymbol{\theta}_j}(\boldsymbol{u}) - \boldsymbol{u}_k) + x_k^j \; \varphi(\boldsymbol{u} - \boldsymbol{u}_k)) \; a_i^j(\boldsymbol{u}) \; \mathrm{d}\boldsymbol{u},
\end{aligned}
$$

where x_k^0 and x_k^j, with $k = 1, \ldots, n$, are the samples describing x_0 and x_j respectively. To facilitate the implementation of this observation model, we assume that the grid G has a sufficiently high resolution so that the integral above is "well" approximated by its Riemann sum on this grid. In the following, we consider that

$$y_{ij} = \sum_{k'=1}^{n} \sum_{k=1}^{n} x_k^0 \, \varphi(\tau_{\boldsymbol{\theta}_j}(\boldsymbol{u}_{k'}) - \boldsymbol{u}_k) a_i^j(\boldsymbol{u}_{k'}) + \sum_{k=1}^{n} x_k^j \, a_i^j(\boldsymbol{u}_k),$$

where we supposed that the pixels of the grid have a unit square area. Therefore, the measurements vector $\boldsymbol{y}_j$ is equal to $\mathsf{A}_j \left(\mathsf{S}(\boldsymbol{\theta}_j) \boldsymbol{x}_0 + \boldsymbol{x}_j \right)$, where $\mathsf{A}_j = (a_i^j(\boldsymbol{u}_k))_{ik} \in \mathbb{R}^{m \times n}$, and $\mathsf{S} \colon \mathbb{R}^q \to \mathbb{R}^{n \times n}$ satisfies

$$\mathsf{S}(\boldsymbol{\theta}_j) = \begin{bmatrix} \varphi\left(\tau_{\boldsymbol{\theta}_j}(\boldsymbol{u}_1) - \boldsymbol{u}_1\right) & \cdots & \varphi\left(\tau_{\boldsymbol{\theta}_j}(\boldsymbol{u}_1) - \boldsymbol{u}_n\right) \\ \vdots & & \vdots \\ \varphi\left(\tau_{\boldsymbol{\theta}_j}(\boldsymbol{u}_n) - \boldsymbol{u}_1\right) & \cdots & \varphi\left(\tau_{\boldsymbol{\theta}_j}(\boldsymbol{u}_n) - \boldsymbol{u}_n\right) \end{bmatrix}. \tag{1.1}$$

Concatenating all the observations in a single vector, we have

$$\begin{bmatrix} \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_l \end{bmatrix} = \begin{bmatrix} \mathsf{A}_1 \mathsf{S}(\boldsymbol{\theta}_1) & \mathsf{A}_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathsf{A}_l \mathsf{S}(\boldsymbol{\theta}_l) & 0 & \cdots & \mathsf{A}_l \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_0 \\ \vdots \\ \boldsymbol{x}_l \end{bmatrix} + \begin{bmatrix} \boldsymbol{n}_1 \\ \vdots \\ \boldsymbol{n}_l \end{bmatrix}, \tag{1.2}$$

where we introduced l vectors $\boldsymbol{n}_1, \ldots, \boldsymbol{n}_l \in \mathbb{R}^m$ to model additive measurement noise. Our goal is now to design a method that estimates the discrete images $\boldsymbol{x} = (\boldsymbol{x}_0^{\mathsf{T}}, \ldots, \boldsymbol{x}_l^{\mathsf{T}})^{\mathsf{T}} \in \mathbb{R}^{(l+1)n}$, and the transformation parameters $\boldsymbol{\theta}^{\mathsf{T}} = (\boldsymbol{\theta}_1^{\mathsf{T}}, \ldots, \boldsymbol{\theta}_l^{\mathsf{T}})^{\mathsf{T}} \in \mathbb{R}^{lq}$, using $\boldsymbol{y} = (\boldsymbol{y}_1^{\mathsf{T}}, \ldots, \boldsymbol{y}_l^{\mathsf{T}})^{\mathsf{T}} \in \mathbb{R}^{lm}$ as sole information.

The inter-correlation between the observed images appears explicitly in the measurement model (1.2). We consider that the observed images share a common signal - the background image - which lives on a low-dimensional manifold. In the literature, other models that exploit the inter-correlation between the measurements have been proposed.

1.2. Related works. The problem of reconstructing an image from multi-view measurements has been studied by several authors in contexts such as compressed sensing, super-resolution, or robust image alignment. We try here to give a brief overview of the different types of techniques used.

When the measurements are obtained by compressed sensing, some techniques, like ours, rely on the estimation of geometric transformations between images. For example, the authors in [13] propose to reconstruct the observed images by assuming that they are sparse in an over-complete dictionary created by scaling, translating and rotating several times a mother waveform. Then, they assume that the decompositions of these images in this dictionary are related by local geometrical transformations. In ?,24, the authors suppose that the images live on a low-dimensional manifold. Then they use a manifold lifting algorithm to obtain a first estimation of the transformation parameters and a standard compressed sensing algorithm for the estimation of the images. An alternate minimization method is then proposed to refine the estimation of the transformation parameters and of the images. However, no proof of

convergence of the method is provided and, to work efficiently, the algorithm requires the use of a specific measurement matrix, implemented with noiselets, to estimate the transformation parameters in a coarse-to-fine scales fashion. We also remark that the presence of occlusions in the images is not considered in this model.

Continuing with compressed sensing, other techniques rely on more general interpolation model between images. In [40, 41], the images are first estimated with a bidirectional interpolation using two known neighbor images and then corrected using the acquired measurements. An iterative process, alternating between interpolation and reconstruction, is then proposed to refine the estimation of the images. Note that a similar idea is used in [2] for dynamic MRI. In [22], a slightly simpler model is used. The authors assume that the difference between two images is sparse in a wavelet basis and that the background image is sparse in the gradient domain. Then, a convex minimization problem is solved to reconstruct jointly all the images. In [15, 19], the interpolation relies on "disparity maps" that indicate the correspondences between the pixels of two images. These maps are used to estimate the images from the measurements, and the newly estimated pictures can be used to refine the disparity maps. The process is then iterated several times. We remark that the authors do not provide any proof of convergence. We note that the authors deal with the problem of occlusions by supposing that they are sparse in image space. Let us also mention that a similar idea is proposed in [39]. Finally, in [34], the authors use a linear dynamical system to model the intercorrelation between frames, and reconstruct a video sequence.

When the matrices $A_1, \ldots, A_m$ implement a blurring and downsampling operator, one can try to obtain a high-resolution background image from the images observed at low-resolution by solving (1.2). Super-resolution from multiple frames has also been widely studied and several image priors and correlation models between images have been proposed to solve this problem. As before, some techniques model the correlation between images using geometric transformations [18, 44] while other ones use more general correlation models [28, 36].

Let us now highlight a similarity between our model and the one used in [25, 26] for robust image alignment, which consists in aligning a set of images despite the presence of large occlusions. With the measurement model (1.2), one can attempt to solve this problem by initializing the vectors $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_l$ with the observed images and the observations matrices $A_1, \ldots, A_l$ with the identity. Then, the algorithm should estimate the background image, the transformation parameters that align this background image on the observed ones, and the foreground images. In [25, 26], a different approach, called RASL, is proposed. The authors concatenate the initial images $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_l$ in a matrix and explain that, after alignment of these images, this matrix can be decomposed as the sum of a low-rank matrix (modeling correlated components) and a sparse matrix (modeling occlusions). This model serves as the criterion to align the images. Note that in the limit where the rank of the final matrix modeling the correlated components is 1, their method, like ours, also decomposes the initial images into a single background image and l foreground images. Finally, let us mention that even though TILT [46] resembles RASL at first sight, their purpose is different. TILT is a method to correct and align low-rank textures and takes as input only one image. This is a different problem than the one studied here.

1.3. Main contribution and organization of the paper. In the present work, we propose

a novel algorithm for *joint* registration and reconstruction of a set of misaligned images from linear measurements and, in contrast to the methods described above, provide a *convergence* proof of our algorithm. Furthermore, the model used permits us to easily deal with the problem of occlusions. It is also general enough to allow us to handle many image reconstruction problems. In order to study the efficiency of our algorithm, we use extensive numerical simulations in several applications. These numerical experiments confirm that our algorithm is competitive with well-known methods for robust image alignment, compressed sensing, and super-resolution.

In Section 2, we identify the solution of the multi-view imaging problem with the global minimizer of a non-convex functional, and propose an algorithm for the minimization of this functional. In Section 3, we study the convergence of the reconstruction method, and prove that the sequence of *estimates* (images and transformation parameters) converges to a critical point of the functional. The efficiency of the algorithm is then demonstrated experimentally in Section 4 for several problems. In the spirit of reproducible research, our code[1] and data needed to reproduce the results presented in this paper can be downloaded openly at `lts2www.epfl.ch/people/gilles/softwares`. We finally conclude in Section 5. Some results required to prove the convergence of the algorithm are gathered in the appendix.

Note that we already briefly studied the problem of reconstructing a set of images from multi-view measurements in [29], where the measurement model (1.2) was also used. However, the technique proposed to reconstruct the images was based on the Bregman iterative regularization procedure, which is different from the one used here. Furthermore, the convergence results presented in [29] essentially concern the decrease of the objective value and, unlike here, do not ensure the convergence of the sequence of estimated images and parameters.

1.4. Notations and definitions. The Euclidean scalar product of $\mathbb{R}^n$ is denoted $\langle \cdot, \cdot \rangle$ and $\| \cdot \|_2$ is the corresponding ℓ_2-norm. The ℓ_1-norm of a vector $\boldsymbol{x} = (x_i)_{1 \leqslant i \leqslant n} \in \mathbb{R}^n$ is defined as $\|\boldsymbol{x}\|_1 = \sum_{i=1}^{n} |x_i|$.

Let C be a non-empty closed convex subset of $\mathbb{R}^n$. The indicator function of C is denoted $i_C \colon \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$. It is a proper lower semicontinuous convex function that satisfies $i_C(\boldsymbol{x}) = 0$ if $\boldsymbol{x} \in C$, and $i_C(\boldsymbol{x}) = +\infty$ otherwise.

Let $f \colon \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a proper lower semicontinuous function. The domain of f is denoted and defined by $\mathrm{dom} f = \{\boldsymbol{x} \in \mathbb{R}^n : f(\boldsymbol{x}) < +\infty\}$. The subdifferential of f at $\boldsymbol{x} \in \mathrm{dom} f$ is denoted by $\partial f(\boldsymbol{x})$ (see, e.g., Section 2 of [3] for the definition). Note that $\partial f(\boldsymbol{x}) = \emptyset$ if $\boldsymbol{x} \notin \mathrm{dom} f$. A necessary (but not sufficient) condition for $\boldsymbol{x}^* \in \mathbb{R}^n$ to be a minimizer of f is $\partial f(\boldsymbol{x}^*) \ni \boldsymbol{0}$. A point that satisfies the last condition is called a critical point of f.

Finally, $\mathcal{C}^k$ denotes the set of functions that are k times continuously differentiable.

2. A non-convex optimization technique. In this section, we first identify the solution of the compressed multi-view imaging problem with the minimizer of a non-convex functional and propose an algorithm to minimize this functional.

2.1. Solution as minimizer of a non-convex functional. Let $\epsilon \geqslant 0$ be an upper bound on the energy of the measurement noise, i.e., $\|\boldsymbol{n}\|_2 \leqslant \epsilon$. Our objective is to find a set of

---

[1]We used the SPARCO toolbox available at `www.cs.ubc.ca/labs/scl/sparco/` [9].

parameters $\boldsymbol{\theta}^*$ and images $\boldsymbol{x}^*$ that satisfy the constraint $\|\mathsf{A}(\boldsymbol{\theta}^*)\,\boldsymbol{x}^* - \boldsymbol{y}\|_2 \leqslant \epsilon$, where

$$\mathsf{A}(\boldsymbol{\theta}) = \begin{bmatrix} \mathsf{A}_1\mathsf{S}(\boldsymbol{\theta}_1) & \mathsf{A}_1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathsf{A}_l\mathsf{S}(\boldsymbol{\theta}_l) & 0 & \ldots & \mathsf{A}_l \end{bmatrix} \in \mathbb{R}^{lm \times (l+1)n}.$$

As infinitely many solutions satisfy this constraint, prior information is needed to restrict the set of admissible solutions. To regularize this ill-posed inverse problem, we can, for example, search for images sparse in a wavelet basis by minimizing the ℓ_1-norm of their wavelet coefficients, or seek a solution that minimizes the Total Variation (TV) norm if the images are piecewise constant [33]. We may also require the transformation parameters $\boldsymbol{\theta}_j$, with $j = 1, \ldots, l$, to belong to some convex sets $\Theta_j = \{\boldsymbol{\theta}_j \in \mathbb{R}^q : \underline{\boldsymbol{\theta}}_j \leqslant \boldsymbol{\theta}_j \leqslant \bar{\boldsymbol{\theta}}_j\}$ where $\underline{\boldsymbol{\theta}}_j \in \mathbb{R}^q$ and $\bar{\boldsymbol{\theta}}_j \in \mathbb{R}^q$ are pre-defined upper and lower bounds[2]. Therefore, an estimate of the images and transformation parameters can be obtained by solving a minimization problem of the form

$$\min_{(\boldsymbol{x},\boldsymbol{\theta})} f(\boldsymbol{x}) + \kappa \,\|\mathsf{A}(\boldsymbol{\theta})\,\boldsymbol{x} - \boldsymbol{y}\|_2^2 \ \text{ subject to } \ \boldsymbol{\theta} \in \Theta, \tag{2.1}$$

where $f \colon \mathbb{R}^{(l+1)n} \to \mathbb{R} \cup \{+\infty\}$ is a proper lower-semicontinuous convex function, $\kappa^{-1} > 0$ is a regularizing parameter that should be adjusted with the noise level ϵ, and $\Theta = \{\boldsymbol{\theta} = (\boldsymbol{\theta}_1^\mathsf{T}, \ldots, \boldsymbol{\theta}_l^\mathsf{T})^\mathsf{T} \in \mathbb{R}^{lq} : \boldsymbol{\theta}_j \in \Theta_j, \ j = 1, \ldots, l\}$.

Problem (2.1) is non-linear and finding a global minimizer is not trivial. Nevertheless, one can still try to solve this problem using an alternating minimization technique such as the block coordinate descent method [42]. There is however no guarantee here that this technique converges to a critical point of Problem (2.1). Recently, Attouch *et al.* proposed in [3] a new type of alternating algorithm for the minimization of a non-convex functional and, more importantly, managed to prove that the sequence generated by the algorithm converges to a critical point of the functional for a wide class of problems. As said by the authors, their algorithm can be interpreted as a proximal regularization of the Gauss-Seidel method where cost-to-move functions are added in the minimization procedure. These convergence results were then generalized to inexact descent methods that satisfy a sufficient decrease condition [4]. Based on this work, we develop a minimization method for problem (2.1) and prove that the generated sequence converges to a critical point $(\boldsymbol{x}^*, \boldsymbol{\theta}^*)$ of the functional $L \colon \mathbb{R}^{(l+1)n} \times \mathbb{R}^{lq} \to \mathbb{R} \cup \{+\infty\}$ defined as

$$L(\boldsymbol{x}, \boldsymbol{\theta}) = f(\boldsymbol{x}) + \kappa \,\|\mathsf{A}(\boldsymbol{\theta})\,\boldsymbol{x} - \boldsymbol{y}\|_2^2 + i_\Theta(\boldsymbol{\theta}). \tag{2.2}$$

Let us recall that a global minimizer of L is also a critical point but that a critical point of L is not necessarily a global minimizer. The point $(\boldsymbol{x}^*, \boldsymbol{\theta}^*)$ might thus only be a local minimizer or a saddle point of Problem (2.1).

Problem (2.1) is non-linear and finding a global minimizer is not trivial. The proposed algorithm is a descent method. It generates a sequence of estimates $(\boldsymbol{x}^k, \boldsymbol{\theta}^k)_{k \in \mathbb{N}}$ such that $L(\boldsymbol{x}^{k+1}, \boldsymbol{\theta}^{k+1}) \leqslant L(\boldsymbol{x}^k, \boldsymbol{\theta}^k)$ for all $k \in \mathbb{N}$, and consists of two steps. We start by describing each of these steps and study the convergence of the algorithm in Section 3.

---

[2]Let $\bar{\boldsymbol{\theta}} = (\bar{\theta}_i)_{1 \leqslant i \leqslant q} \in \mathbb{R}^q$, $\boldsymbol{\theta} = (\theta_i)_{1 \leqslant i \leqslant q} \in \mathbb{R}^q$, $\boldsymbol{\theta} \leqslant \bar{\boldsymbol{\theta}}$ means that $\theta_i \leqslant \bar{\theta}_i$ for all $i \in \{1, \ldots, q\}$.

2.2. First step of the algorithm. Let $(\boldsymbol{x}^k, \boldsymbol{\theta}^k) \in \mathbb{R}^{(l+1)n} \times \Theta$ be the estimates obtained after k iterations of the algorithm. The first step consists in finding a new estimate $\boldsymbol{x}^{k+1} \in \mathbb{R}^{(l+1)n}$ that decreases the value of the objective function L while keeping $\boldsymbol{\theta}^k$ fixed. Let us choose $\boldsymbol{x}^{k+1}$ as a solution of

$$\min_{\boldsymbol{x} \in \mathbb{R}^{(l+1)n}} f(\boldsymbol{x}) + \kappa \, \|\mathsf{A}(\boldsymbol{\theta}^k)\, \boldsymbol{x} - \boldsymbol{y}\|_2^2 + \frac{\lambda_{\boldsymbol{x}}}{2} \, g(\boldsymbol{x} - \boldsymbol{x}^k), \tag{2.3}$$

where $\lambda_{\boldsymbol{x}} > 0$, and $g \colon \mathbb{R}^{(l+1)n} \to \mathbb{R}$ is proper lower-semicontinuous convex function such that $g(\boldsymbol{x}) \geqslant 0$ for all $\boldsymbol{x} \in \mathbb{R}^{(l+1)n}$ and $g(\boldsymbol{0}) = 0$. It is clear that (2.3) implies

$$L(\boldsymbol{x}^{k+1}, \boldsymbol{\theta}^k) + \frac{\lambda_{\boldsymbol{x}}}{2} \, g(\boldsymbol{x}^{k+1} - \boldsymbol{x}^k) \leqslant L(\boldsymbol{x}^k, \boldsymbol{\theta}^k). \tag{2.4}$$

Hence $L(\boldsymbol{x}^{k+1}, \boldsymbol{\theta}^k) \leqslant L(\boldsymbol{x}^k, \boldsymbol{\theta}^k)$ with a decrease of $\lambda_{\boldsymbol{x}} \, g(\boldsymbol{x}^{k+1} - \boldsymbol{x}^k)/2$, at least. Note that this first minimization problem is convex in $\boldsymbol{x}$ and can be solved using, e.g., the algorithms presented in [8,14,32].

In Problem (2.3), the function g acts as a proximal term and $\lambda_{\boldsymbol{x}}$ as a stepsize. It ensures a sufficient decrease of the functional L at each iteration and is essential to prove the convergence of the sequence $(\boldsymbol{x}^k, \boldsymbol{\theta}^k)_{k \in \mathbb{N}}$ to a critical point. This function also provides, to some extent, a control on the evolution of the generated sequence, and we realized that it must be chosen carefully to achieve good and stable results in all the scenarios tested. Note that in the work [3] of Attouch *et al.*, the function g is a squared ℓ_2-norm.

From the multi-view measurements $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_l$, the algorithm should reconstruct the background image, separate the occlusions (foreground), and provide the transformation parameters that register the initial images between them. This problem is solved here by alternatively refining the estimation of the images $\boldsymbol{x}$ (background and foreground) and of the transformation parameters $\boldsymbol{\theta}$ (image registration). We noticed that to improve the quality of the registration, it is better to have a reconstruction of the images in a coarse-to-fine scales fashion, as in [24,45]. In these papers, the authors use a dedicated measurement matrix that allows this type of reconstruction. In the current paper, this goal is achieved by choosing a function g that favors the reconstruction of the coarse scales first and then of the fine scales. There is thus no constraint on the choice of the measurement matrices anymore.

In order to have a coarse-to-fine scales reconstruction of the images, we use a wavelet tight-frame $\mathsf{W} \in \mathbb{R}^{n \times p}$, with $p \geqslant n$, i.e., $\mathsf{W}\mathsf{W}^{\mathsf{T}}$ is equal to $\mathsf{I}_n \in \mathbb{R}^{n \times n}$, the identity matrix. One should remark that most of the largest wavelet coefficients of a natural image live at the coarsest scales and that the amplitude of these coefficients decreases when going to finer scales (except at the few singularities). To achieve our goal, we need a function g that favors a reconstruction of the coarse scales first, by selecting the biggest wavelet coefficients, and then of the fine scales, by selecting the smallest wavelet coefficients. Let $\Psi \in \mathbb{R}^{(l+1)n \times (l+1)p}$ be the block-diagonal matrix built by repeating $l + 1$ times the matrix W on the diagonal. We have $\Psi\Psi^{\mathsf{T}} = \mathsf{I}_{(l+1)n}$. Take $\boldsymbol{x}^{k+1}$ as a minimizer of $f(\boldsymbol{x}) + \kappa \, \|\mathsf{A}(\boldsymbol{\theta}^k)\, \boldsymbol{x} - \boldsymbol{y}\|_2^2 + \lambda_{\boldsymbol{x}}\|\Psi^{\mathsf{T}}(\boldsymbol{x} - \boldsymbol{x}^k)\|_1/2$, and remember that the ℓ_1-norm favors solutions with few large coefficients. Hence, the function f forces $\boldsymbol{x}^{k+1}$ to fit our prior, the quadratic term forces $\boldsymbol{x}^{k+1}$ to be consistent with the observations, and the cost-to-move function $\boldsymbol{x} \mapsto \|\Psi^{\mathsf{T}}(\boldsymbol{x} - \boldsymbol{x}^k)\|_1$ imposes that $\boldsymbol{x}^{k+1}$ differs from $\boldsymbol{x}^k$ by only a few

large wavelet coefficients. Starting from $\boldsymbol{x}^0 = \boldsymbol{0} \in \mathbb{R}^{(l+1)n}$ at $k = 0$, the first estimated images $\boldsymbol{x}^1$ have thus a sparse decomposition in the wavelet frame W. The sparsity increases with the parameter $\lambda_{\boldsymbol{x}}$. As we are reconstructing natural images, these few large coefficients mainly live at large scales and $\boldsymbol{x}^1$ is a coarse approximation of the solution. The estimation of the images is then refined at the next iteration, and the cost-to-move function favors the selection of the next biggest wavelet coefficients living at finer scales. Thanks to this procedure, we have a coarse-to-fine scales reconstruction of the images. This behavior will be illustrated in Section 4.

Note that to be able to prove the convergence of the sequence $(\boldsymbol{x}^k, \boldsymbol{\theta}^k)_{k \in \mathbb{N}}$ to a critical point of L with the results presented in [3, 4], we actually substitute the Huber function for the ℓ_1-norm as cost-to-move function. The Huber function is a smooth approximation of the ℓ_1-norm that depends on a smoothing parameter $\mu > 0$. This parameter can be chosen small so that both functions are nearly identical. Let $\boldsymbol{\alpha} = (\alpha_i)_{1 \leqslant i \leqslant p} \in \mathbb{R}^{(l+1)p}$, the Huber function $h_\mu : \mathbb{R}^{(l+1)p} \to \mathbb{R}$ satisfies

$$h_\mu(\boldsymbol{\alpha}) = \sum_{i=1}^{(l+1)p} h_i, \quad \text{where} \quad h_i = \begin{cases} \alpha_i^2/(2\mu), & \text{if } |\alpha_i| < \mu, \\ |\alpha_i| + \mu/2, & \text{otherwise,} \end{cases} \quad \forall i \in \{1, \dots, (l+1)p\}.$$

From now on, the term $g(\boldsymbol{x} - \boldsymbol{x}^k)$ in Problem (2.3) is replaced by $h_\mu(\Psi^\mathsf{T}(\boldsymbol{x} - \boldsymbol{x}^k))$ (including in our implementation of the algorithm).

2.3. Second step of the algorithm. The second step consists in updating the transformation parameters to further decrease the value of the objective function L. As the functions $\boldsymbol{\theta} \mapsto \|\mathsf{A}(\boldsymbol{\theta}^k)\,\boldsymbol{x} - \boldsymbol{y}\|_2^2$ and i_Θ are separable in $\boldsymbol{\theta}_j$, with $j = 1, \dots, l$, we optimize the transformation parameters separately for each observation.

Let $(\boldsymbol{x}^k, \boldsymbol{\theta}^k) \in \mathbb{R}^{(l+1)n} \times \Theta$ be the estimates obtained after k iterations of the algorithm and $\boldsymbol{x}^{k+1} \in \mathbb{R}^{(l+1)n}$ be the solution of problem (2.3). To simplify the notations, we introduce new functions $Q_j^{k+1} : \mathbb{R}^q \to \mathbb{R}$, with $j = 1, \dots, l$, defined as

$$Q_j^{k+1}(\boldsymbol{\theta}_j) = \|\mathsf{A}_j \mathsf{S}(\boldsymbol{\theta}_j) \boldsymbol{x}_0^{k+1} + \mathsf{A}_j \boldsymbol{x}_j^{k+1} - \boldsymbol{y}_j\|_2^2. \tag{2.5}$$

Note that $\|\mathsf{A}(\boldsymbol{\theta})\boldsymbol{x}^{k+1} - \boldsymbol{y}\|_2^2 = \sum_{j=1}^l Q_j^{k+1}(\boldsymbol{\theta}_j)$ and that $i_\Theta(\boldsymbol{\theta}) = \sum_{j=1}^l i_{\Theta_j}(\boldsymbol{\theta}_j)$. We choose to update the transformation parameters with the following projected Newton-like method [10, 35].

Let us assume that the entries of the matrix $\mathsf{S}(\boldsymbol{\theta}_j)$ are differentiable with respect to the transformation parameters. The first order Taylor expansion of $\mathsf{S}(\boldsymbol{\theta}_j)\boldsymbol{x}_0^{k+1}$ at $\boldsymbol{\theta}_i^k$ is $\mathsf{S}(\boldsymbol{\theta}_i^k)\boldsymbol{x}_0^{k+1} + \mathsf{J}(\boldsymbol{\theta}_j^k)(\boldsymbol{\theta}_j - \boldsymbol{\theta}_j^k)$ with

$$\mathsf{J}(\boldsymbol{\theta}_j^k) = \left(\partial_{\theta_{1j}} \mathsf{S}(\boldsymbol{\theta}_j^k)\,\boldsymbol{x}_0^{k+1}\ ,\ \dots\ ,\ \partial_{\theta_{qj}} \mathsf{S}(\boldsymbol{\theta}_j^k)\,\boldsymbol{x}_0^{k+1} \right) \in \mathbb{R}^{n \times q}.$$

Therefore,

$$Q_j^{k+1}(\boldsymbol{\theta}_j^k) + \left\langle \nabla Q_j^{k+1}(\boldsymbol{\theta}_j^k), \boldsymbol{\theta}_j - \boldsymbol{\theta}_j^k \right\rangle + \|\mathsf{A}_j \mathsf{J}(\boldsymbol{\theta}_j^k)(\boldsymbol{\theta}_j - \boldsymbol{\theta}_j^k)\|_2^2,$$

with

$$\nabla Q_j^{k+1}(\boldsymbol{\theta}_j^k) = 2 \left(\mathsf{A}_j \mathsf{J}(\boldsymbol{\theta}_j^k) \right)^\top \left(\mathsf{A}_j \mathsf{S}(\boldsymbol{\theta}_j^k) \boldsymbol{x}_0^{k+1} + \mathsf{A}_j \boldsymbol{x}_j^{k+1} - \boldsymbol{y}_j \right),$$

is a second order approximation of Q_j^{k+1} at $\boldsymbol{\theta}_j^k$. The positive semi-definite matrix

$$\mathsf{H}(\boldsymbol{\theta}_j^k) = 2 \left(\mathsf{A}_j \mathsf{J}(\boldsymbol{\theta}_j^k) \right)^\top \left(\mathsf{A}_j \mathsf{J}(\boldsymbol{\theta}_j^k) \right), \tag{2.6}$$

can be viewed as an approximation of the Hessian of the function Q_j^{k+1}. To update the transformation parameters, we choose to minimize this quadratic approximation to which we add another quadratic term that ensures a decrease of the objective function. We take as next estimate of the transformation parameters

$$\boldsymbol{\theta}_j^{k+1} = \operatorname*{argmin}_{\boldsymbol{\theta}_j \in \Theta_j} \ \left\langle \nabla Q_j^{k+1}(\boldsymbol{\theta}_j^k), \boldsymbol{\theta}_j - \boldsymbol{\theta}_j^k \right\rangle + \frac{1}{2} \left\langle \boldsymbol{\theta}_j - \boldsymbol{\theta}_j^k, \left[\mathsf{H}(\boldsymbol{\theta}_j^k) + 2^i \lambda_{\boldsymbol{\theta}} \mathsf{I}_q \right] (\boldsymbol{\theta}_j - \boldsymbol{\theta}_j^k) \right\rangle, \tag{2.7}$$

where $\mathsf{I}_q \in \mathbb{R}^{q \times q}$ is the identity matrix, $\lambda_{\boldsymbol{\theta}} > 0$, and i is the smallest positive integer such that

$$Q_j^{k+1}(\boldsymbol{\theta}_j^{k+1}) \leqslant Q_j^{k+1}(\boldsymbol{\theta}_j^k) + \left\langle \nabla Q_j^{k+1}(\boldsymbol{\theta}_j^k), \boldsymbol{\theta}_j^{k+1} - \boldsymbol{\theta}_j^k \right\rangle$$
$$+ \ \frac{1}{2} \left\langle \boldsymbol{\theta}_j^{k+1} - \boldsymbol{\theta}_j^k, \left[\mathsf{H}(\boldsymbol{\theta}_j^k) + (2^i - 1) \lambda_{\boldsymbol{\theta}} \mathsf{I}_q \right] (\boldsymbol{\theta}_j^{k+1} - \boldsymbol{\theta}_j^k) \right\rangle. \tag{2.8}$$

The existence of such an i is discussed in the next section[3]. One can remark that if the minimization (2.7) was performed over all $\mathbb{R}^q$ instead of Θ_j, the solution would be $\boldsymbol{\theta}_j^{k+1} = \boldsymbol{\theta}_j^k - [\mathsf{H}(\boldsymbol{\theta}_j^k) + 2^i \lambda_{\boldsymbol{\theta}} \mathsf{I}_q]^{-1} \nabla Q_j^{k+1}(\boldsymbol{\theta}_j^k)$, as in a Newton method with Hessian $\mathsf{H}(\boldsymbol{\theta}_j^k) + 2^i \lambda_{\boldsymbol{\theta}} \mathsf{I}_q$. One can also note that if we had chosen $\mathsf{H}(\boldsymbol{\theta}_j^k) = \mathbf{0}$ then $\boldsymbol{\theta}_j^{k+1}$ would be obtained with a simple projected gradient update. We noticed however that the Newton-like update (2.7), which can, for example, be solved using the algorithms presented in [8], yields more accurate results.

To check that our choice of new transformation parameters decreases the value of the objective function L, one just has to combine (2.7) and (2.8) to realize that

$$Q_j^{k+1}(\boldsymbol{\theta}_j^{k+1}) + \frac{\lambda_{\boldsymbol{\theta}}}{2} \|\boldsymbol{\theta}_j^{k+1} - \boldsymbol{\theta}_j^k\|_2^2 \ \leqslant$$
$$Q_j^{k+1}(\boldsymbol{\theta}_j^k) + \left\langle \nabla Q_j^{k+1}(\boldsymbol{\theta}_j^k), \boldsymbol{\theta}_j - \boldsymbol{\theta}_j^k \right\rangle + \frac{1}{2} \left\langle \boldsymbol{\theta}_j - \boldsymbol{\theta}_j^k, \left[\mathsf{H}(\boldsymbol{\theta}_j^k) + 2^i \lambda_{\boldsymbol{\theta}} \mathsf{I}_q \right] (\boldsymbol{\theta}_j - \boldsymbol{\theta}_j^k) \right\rangle,$$

for any $\boldsymbol{\theta}_j \in \Theta_j$. Choosing $\boldsymbol{\theta}_j = \boldsymbol{\theta}_j^k$ in the last inequality, multiplying it by κ, and summing all the inequalities obtained for j from 1 to l yields

$$L(\boldsymbol{x}^{k+1}, \boldsymbol{\theta}^{k+1}) + \frac{\kappa \lambda_{\boldsymbol{\theta}}}{2} \|\boldsymbol{\theta}^{k+1} - \boldsymbol{\theta}^k\|_2^2 \leqslant L(\boldsymbol{x}^{k+1}, \boldsymbol{\theta}^k). \tag{2.9}$$

Therefore $L(\boldsymbol{x}^{k+1}, \boldsymbol{\theta}^{k+1}) \leqslant L(\boldsymbol{x}^{k+1}, \boldsymbol{\theta}^k)$ with a decrease of $\kappa \lambda_{\boldsymbol{\theta}} \|\boldsymbol{\theta}^{k+1} - \boldsymbol{\theta}^k\|_2^2 / 2$, at least.

---

[3]Note that if one is able to compute the Lipschitz constant Λ_j^{k+1} of ∇Q_j^{k+1}, then Λ_j^{k+1} can be substituted for $2^i \lambda_{\boldsymbol{\theta}}$ and the algorithm also converges.

3. Convergence analysis. In this section, we analyze the convergence of the optimization method described in Section 2. We first present sufficient conditions under which the generated sequence converges to a critical point of L. Then, we describe several cases where these conditions are satisfied. In particular, we study different types of geometric transformations which meet these requirements. Finally, we discuss the influence of the cost-to-move parameters on the final reconstruction quality.

Algorithm 1

Inputs: measurements $\boldsymbol{y} \in \mathbb{R}^{lm}$, wavelet tight frame $\Psi \in \mathbb{R}^{(l+1)n \times (l+1)p}$, regularization parameter $\kappa > 0$, parameters $(\lambda_{\boldsymbol{x}}^k)_{k \in \mathbb{N}}, \lambda_{\boldsymbol{\theta}} > 0$, and bounds $\bar{\boldsymbol{\theta}} \geqslant \underline{\boldsymbol{\theta}}$.

Initializations: set $k = 0$, $\boldsymbol{x}^0 = \mathbf{0} \in \mathbb{R}^{(l+1)n}$, and $\boldsymbol{\theta}^0 \in \Theta$.

repeat

  1) Set

$$\boldsymbol{x}^{k+1} \leftarrow \underset{\boldsymbol{x} \in \mathbb{R}^{(l+1)n}}{\operatorname{argmin}} \; L(\boldsymbol{x}, \boldsymbol{\theta}^k) + \frac{\lambda_{\boldsymbol{x}}^k}{2} \, h_\mu \left(\Psi^\mathsf{T}(\boldsymbol{x} - \boldsymbol{x}^k) \right).$$

  2) For all $j = 1, \ldots, l$, set

$$\boldsymbol{\theta}_j^{k+1} \leftarrow \underset{\boldsymbol{\theta}_j \in \Theta_j}{\operatorname{argmin}} \; \left\langle \nabla Q_j^{k+1}(\boldsymbol{\theta}_j^k), \boldsymbol{\theta}_j - \boldsymbol{\theta}_j^k \right\rangle + \frac{1}{2} \left\langle \boldsymbol{\theta}_j - \boldsymbol{\theta}_j^k, \left[\mathsf{H}(\boldsymbol{\theta}_j^k) + 2^i \lambda_{\boldsymbol{\theta}} \mathsf{I} \right] (\boldsymbol{\theta}_j - \boldsymbol{\theta}_j^k) \right\rangle,$$

  where Q_j^{k+1} is defined in (2.5), $\mathsf{H}(\boldsymbol{\theta}_j^k)$ is defined in (2.6), and i is the smallest positive integer such that (2.8) is satisfied.

  3) $k \leftarrow k + 1$

until convergence or $k \geqslant k_{\max}$

Outputs: Estimated images $\boldsymbol{x}^*$ and transformation parameters $\boldsymbol{\theta}^*$.

3.1. General convergence result. Let us consider Algorithm 1 which is a summary of the optimization method described in the previous section. We are now in position to state our main convergence result.

Theorem 3.1. *Let L be the objective function defined in (2.2) with $\kappa > 0$. Assume that L is bounded below, that the entries of the matrix S defined in (1.1) are twice continuously differentiable with respect to the transformation parameters, that $\Psi \in \mathbb{R}^{(l+1)n \times (l+1)p}$ satisfies $\Psi \Psi^\mathsf{T} = \mathsf{I}_{(l+1)n}$, and that $0 < \underline{\lambda} \leqslant \lambda_{\boldsymbol{x}}^k, \lambda_{\boldsymbol{\theta}} \leqslant \bar{\lambda}$ for all $k \in \mathbb{N}$. Then, the sequence of estimates $(\boldsymbol{x}^k, \boldsymbol{\theta}^k)_{k \in \mathbb{N}}$ generated by Algorithm 1 is correctly defined and the following statements hold.*

  1. For all $k \geqslant 0$,

$$L(\boldsymbol{x}^{k+1}, \boldsymbol{\theta}^{k+1}) + \frac{\lambda}{2} \left[\kappa \, \|\boldsymbol{\theta}^{k+1} - \boldsymbol{\theta}^k\|_2^2 + h_\mu(\Psi^\mathsf{T}(\boldsymbol{x}^{k+1} - \boldsymbol{x}^k)) \right] \leqslant L(\boldsymbol{x}^k, \boldsymbol{\theta}^k). \qquad (3.1)$$

  Hence $L(\boldsymbol{x}^k, \boldsymbol{\theta}^k)$ does not increase.

2. *The sequences $(\boldsymbol{x}^{k+1} - \boldsymbol{x}^k)_{k\in\mathbb{N}}$ and $(\boldsymbol{\theta}^{k+1} - \boldsymbol{\theta}^k)_{k\in\mathbb{N}}$ converge. Indeed,*

$$\lim_{k\to+\infty} \|\boldsymbol{x}^{k+1} - \boldsymbol{x}^k\|_2 + \|\boldsymbol{\theta}^{k+1} - \boldsymbol{\theta}^k\|_2 = 0. \tag{3.2}$$

3. *Assume that L has the Kurdyca-Łojasiewicz property (see Definition 3.2 in [3]). Then, if the sequence $(\boldsymbol{x}^k)_{k\in\mathbb{N}}$ is bounded, the sequence $(\boldsymbol{x}^k, \boldsymbol{\theta}^k)_{k\in\mathbb{N}}$ converges to a critical point $(\boldsymbol{x}^*, \boldsymbol{\theta}^*)$ of L.*

Proof. Let us start by showing that the sequence generated by Algorithm 1 is well-defined. For the first step of the algorithm, note that, for all $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{\theta}}) \in \mathbb{R}^{(l+1)n} \times \mathbb{R}^{lq}$ and $\tilde{\lambda} > 0$, $\boldsymbol{x} \mapsto L(\boldsymbol{x}, \tilde{\boldsymbol{\theta}})$ is a proper lower semicontinuous convex function which is bounded below and that $\boldsymbol{x} \mapsto \tilde{\lambda}\, h_\mu(\Psi^\mathsf{T}(\boldsymbol{x} - \tilde{\boldsymbol{x}}))/2$ is a proper lower semicontinuous convex function which is coercive. Therefore, the function $\boldsymbol{x} \mapsto L(\boldsymbol{x}, \tilde{\boldsymbol{\theta}}) + \tilde{\lambda}\, h_\mu(\Psi^\mathsf{T}(\boldsymbol{x} - \tilde{\boldsymbol{x}}))/2$ has a minimizer (Corollary 11.15, [7]). For the second step of the algorithm, remark that, for all $(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{x}}_j) \in \mathbb{R}^n \times \mathbb{R}^n$, the function $Q_j \colon \boldsymbol{\theta}_j \mapsto \|\mathsf{A}_j \mathsf{S}(\boldsymbol{\theta}_j)\tilde{\boldsymbol{x}}_0 + \mathsf{A}_j \tilde{\boldsymbol{x}}_j - \boldsymbol{y}_j\|_2^2$ is $\mathcal{C}^1$ with Lipschitz continuous gradient on the closed and bounded convex set Θ_j. Let Λ_j be this Lipschitz constant, whose value *a priori* depends on $(\tilde{\boldsymbol{x}}_0, \tilde{\boldsymbol{x}}_j)$. Using the descent lemma (Lemma 3.1, [4]), we have

$$Q_j(\boldsymbol{\theta}_j^1) \leqslant Q_j(\boldsymbol{\theta}_j^2) + \langle \nabla Q_j(\boldsymbol{\theta}_j^2), \boldsymbol{\theta}_j^1 - \boldsymbol{\theta}_j^2 \rangle + \frac{\Lambda_j}{2} \|\boldsymbol{\theta}_j^1 - \boldsymbol{\theta}_j^2\|_2^2,$$

for any two points $(\boldsymbol{\theta}_j^1, \boldsymbol{\theta}_j^2) \in \Theta_j^2$. This proves the existence of an integer i such that (2.8) is satisfied at each second step of Algorithm 1. The existence of a minimizer in Problem (2.7) is then proved by using, e.g., Proposition 11.14 in [7]. An induction finally shows that sequence $(\boldsymbol{x}^k, \boldsymbol{\theta}^k)_{k\in\mathbb{N}}$ is well-defined.

Inequality (3.1) follows by combination of (2.4) and (2.9) and by using the fact that $\lambda_{\boldsymbol{x}}^k, \lambda_{\boldsymbol{\theta}} \geqslant \underline{\lambda}$ for all $k \in \mathbb{N}$. Then, summing this inequality from $k=0$ to K yields

$$\frac{\underline{\lambda}}{2} \sum_{k=0}^{K} \kappa\, \|\boldsymbol{\theta}^{k+1} - \boldsymbol{\theta}^k\|_2^2 + h_\mu(\Psi^\mathsf{T}(\boldsymbol{x}^{k+1} - \boldsymbol{x}^k)) \leqslant L(\boldsymbol{x}^0, \boldsymbol{\theta}^0) - L(\boldsymbol{x}^{K+1}, \boldsymbol{\theta}^{K+1})$$

As L is bounded below and $L(\boldsymbol{x}^0, \boldsymbol{\theta}^0) = \kappa\|\boldsymbol{y}\|_2^2$, it is clear that the righthand side of the previous inequality is also bounded. Consequently,

$$\sum_{k=0}^{+\infty} h_\mu(\Psi^\mathsf{T}(\boldsymbol{x}^{k+1} - \boldsymbol{x}^k)) + \kappa\, \|\boldsymbol{\theta}^{k+1} - \boldsymbol{\theta}^k\|_2^2 < +\infty,$$

and, using the definition of h_μ and the fact that $\Psi\Psi^\mathsf{T} = \mathsf{I}_{(l+1)n}$, the second point of the theorem holds.

The proof of the third point make use of results established by Attouch *et al.* in [4] and can be found in Appendix A. ■

Note that to study the convergence of Algorithm 1, we considered, for simplicity, that the minimization problems (2.3) and (2.7) can be minimized exactly. However, the results presented by Attouch *et al.* in [4] are very general. The authors prove the convergence of many algorithms even when exact minimizations is not performed. This suggests that Algorithm 1 also convergences with inexact minimizations, upon the introduction of few additional requirements on the "quality" of the minimizations.

The third point of Theorem 3.1 applies if L has the Kurdyca-Łojasiewicz property. As explained in [3], this property is satisfied by several types of functions including semi-algebraic ones. A function $g : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is semi-algebraic if its graph $\{(\boldsymbol{x}, t) \in \mathbb{R}^n \times \mathbb{R} : g(\boldsymbol{x}) = t\}$ is a semi-algebraic subset of $\mathbb{R}^{n+1}$, i.e., if it can be written as a finite union of sets of the form $\{\boldsymbol{x} \in \mathbb{R}^{n+1} : p_i(\boldsymbol{x}) = 0, \ q_i(\boldsymbol{x}) < 0, \ i = 1, \dots, r\}$, where p_i and q_i are polynomials. The ℓ_1-norm is thus an example of such a function. Note that the indicator function of a semi-algebraic set is semi-algebraic, and that the set of semi-algebraic functions is stable under basics operations such as sums, products or compositions (Section 4.3, [3]).

In the definition of the objective function L, the set Θ is semi-algebraic. Consequently, L satisfies the Kurdyca-Łojasiewicz property if, e.g., f and the fidelity term $(\boldsymbol{\theta}, \boldsymbol{x}) \mapsto \|\mathsf{A}(\boldsymbol{\theta})\boldsymbol{x} - \boldsymbol{y}\|_2^2$ are semi-algebraic. For f, one can use, for example, $f(\boldsymbol{x}) = \|\Phi\boldsymbol{x}\|_1$ with any real matrix Φ. This choice ensures that L is bounded below. Also note that if Φ has full column rank, then, using condition (3.1), one can show that the sequence $(\boldsymbol{x}_k)_{k \in \mathbb{N}}$ is bounded. For the fidelity term, its semi-algebraicity depends on the properties of the generating function φ and the type of geometric transformations. In the next section, we present examples of functions φ and geometric transformations such that this term is semi-algebraic and such that the entries of the matrix S are twice continuously differentiable, as required by Theorem 3.1.

3.2. Examples of admissible geometric transformations.

Let us start by studying the simple case of translations. This type of transformation can be represented by 2 parameters $\boldsymbol{\theta} = (\theta_1, \theta_2)^{\mathsf{T}} \in \mathbb{R}^2$ and a function $\tau_{\boldsymbol{\theta}}$ defined as

$$\tau_{\boldsymbol{\theta}}(\boldsymbol{u}) = (u_1 + \theta_1, \ u_2 + \theta_2)^{\mathsf{T}}.$$

Combined, for example, with the cubic interpolator defined in [20] as generating function $\varphi(\boldsymbol{u}) = \phi(u_1)\phi(u_2)$ where

$$\phi(u) = \begin{cases} \frac{3}{2}|u|^3 - \frac{5}{2}|u|^2 + 1, & \text{if } 0 \leqslant |u| < 1, \\ -\frac{1}{2}|u|^3 + \frac{5}{2}|u|^2 - 4|u| + 2, & \text{if } 1 \leqslant |u| < 2, \\ 0, & \text{otherwise,} \end{cases} \tag{3.3}$$

the entries of the matrix S are twice continuously differentiable with respect to the transformation parameters, and the function $(\boldsymbol{\theta}, \boldsymbol{x}) \mapsto \|\mathsf{A}(\boldsymbol{\theta})\boldsymbol{x} - \boldsymbol{y}\|_2^2$ is semi-algebraic. Theorem 3.1 thus applies in this situation. Remark that with this choice of generating function the matrix S is sparse, which allows us to have a fast implementation of the geometric transformations. Note that, instead of the cubic interpolator, we could also have chosen any B-spline interpolators of degree larger than 3 [43].

Then, with the same choice of generating function, we can also handle the case of affine transformations. Indeed, such transformations can be represented by 6 parameters $\boldsymbol{\theta} = (\theta_1, \dots, \theta_6)^{\mathsf{T}} \in \mathbb{R}^6$ with

$$\tau_{\boldsymbol{\theta}}(\boldsymbol{u}) = (\theta_1 u_1 + \theta_2 u_2 + \theta_3, \ \theta_4 u_1 + \theta_5 u_2 + \theta_6)^{\mathsf{T}}, \tag{3.4}$$

and the requirements to apply Theorem 3.1 are also met.

Next, let us show how to handle the case of homographies. An homography is usually represented using 8 parameters $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots \boldsymbol{\theta}_8)^\mathsf{T} \in \mathbb{R}^8$ with $\tau_{\boldsymbol{\theta}}$ satisfying

$$\tau_{\boldsymbol{\theta}}(\boldsymbol{u}) = \left(\frac{\theta_1 u_1 + \theta_2 u_2 + \theta_3}{\theta_7 u_1 + \theta_8 u_2 + 1}, \; \frac{\theta_4 u_1 + \theta_5 u_2 + \theta_6}{\theta_7 u_1 + \theta_8 u_2 + 1} \right)^\mathsf{T}.$$

Unfortunately, the function $(\boldsymbol{\theta}, \boldsymbol{x}) \mapsto \|\mathsf{A}(\boldsymbol{\theta})\boldsymbol{x} - \boldsymbol{y}\|_2^2$ is not semi-algebraic in this case. However, if $|\theta_7 u_1 + \theta_8 u_2| \ll 1$, the transformation function can be approximated by

$$\tau_{\boldsymbol{\theta}}(\boldsymbol{u}) = \begin{pmatrix} \theta_1 u_1 + \theta_2 u_2 + \theta_3 \\ \theta_4 u_1 + \theta_5 u_2 + \theta_6 \end{pmatrix} (1 - \theta_7 u_1 - \theta_8 u_2), \tag{3.5}$$

using a first order Taylor expansion. With this approximation, all the conditions are now fulfilled to apply Theorem 3.1. Note that the condition $|\theta_7 \boldsymbol{u}_1 + \theta_8 \boldsymbol{u}_2| \ll 1$ can be enforced during the reconstruction by using appropriate bounds $\bar{\boldsymbol{\theta}}$ and $\underline{\boldsymbol{\theta}}$.

Beyond these examples, let us mention that if the generating function φ and the transformation function $\tau_{\boldsymbol{\theta}}$ are twice continuously differentiable, Algorithm 1 can also be applied and the first two points of Theorem 3.1 still hold.

3.3. Influence of the algorithm parameters. Let us highlight that, even though the sequence $(\boldsymbol{x}^k, \boldsymbol{\theta}^k)_{k \in \mathbb{N}}$ generated by the algorithm converges to a critical point of L for all strictly positive parameter $\lambda_{\boldsymbol{\theta}}$ and strictly positive bounded sequence $(\lambda_{\boldsymbol{x}}^k)_{k \in \mathbb{N}}$, different values of these parameters yield different results. The choice of these parameters is thus important.

For the sequence $(\lambda_{\boldsymbol{x}}^k)_{k \in \mathbb{N}}$, we start from large values and thus constrain $\boldsymbol{x}^{k+1}$ to stay "close" to $\boldsymbol{x}^k$ when the estimated images are still inaccurate. Starting with large values is also essential to obtain a first estimated images made of wavelets at coarse scales only. As a rule of thumb, we have noticed that the best reconstructions were obtained when $\lambda_{\boldsymbol{x}}^0$ is chosen such that $\|\mathsf{A}(\boldsymbol{\theta}^0)\boldsymbol{x}^1 - \boldsymbol{y}\|_2^2 / \|\boldsymbol{y}\|_2^2 \in [0.1, 0.2]$, after the first step of the first iteration. Then, we slightly decrease the value of $\lambda_{\boldsymbol{x}}^k$ at each iteration with the recursion rule $\lambda_{\boldsymbol{x}}^{k+1} = \max(0.9\lambda_{\boldsymbol{x}}^k, 0.1)$. The parameter $\lambda_{\boldsymbol{\theta}}$ applying on the transformation parameters seems to have less influence on the final reconstruction quality, and we keep it fixed at 0.1 in all the experiments of the next section.

Finally, concerning the bounds $\underline{\boldsymbol{\theta}}$ and $\bar{\boldsymbol{\theta}}$ constraining the transformation parameters, we simply choose them large enough to ensure that the actual transformation parameters between images are in the allowed range, while making sure that uninteresting solutions, such as translations larger than the size of the images or scaling parameters at zero, are prevented.

3.4. Limitations of our algorithm. We now conclude this section by discussing some limitations of our algorithm.

A first obvious limitation of our technique arises when the geometric transformations of the background image between two different observations are not global but local. This situation occurs, for example, when the observed scene is not planar. In this case, the geometric transformations differ from one pixel location to another and depend on the depth of the observed scene. Note however that the following two properties of the proposed method may help to compensate for such an issue. First, one should note that the presence of foreground images in the measurement model can render our method robust to a small transformation
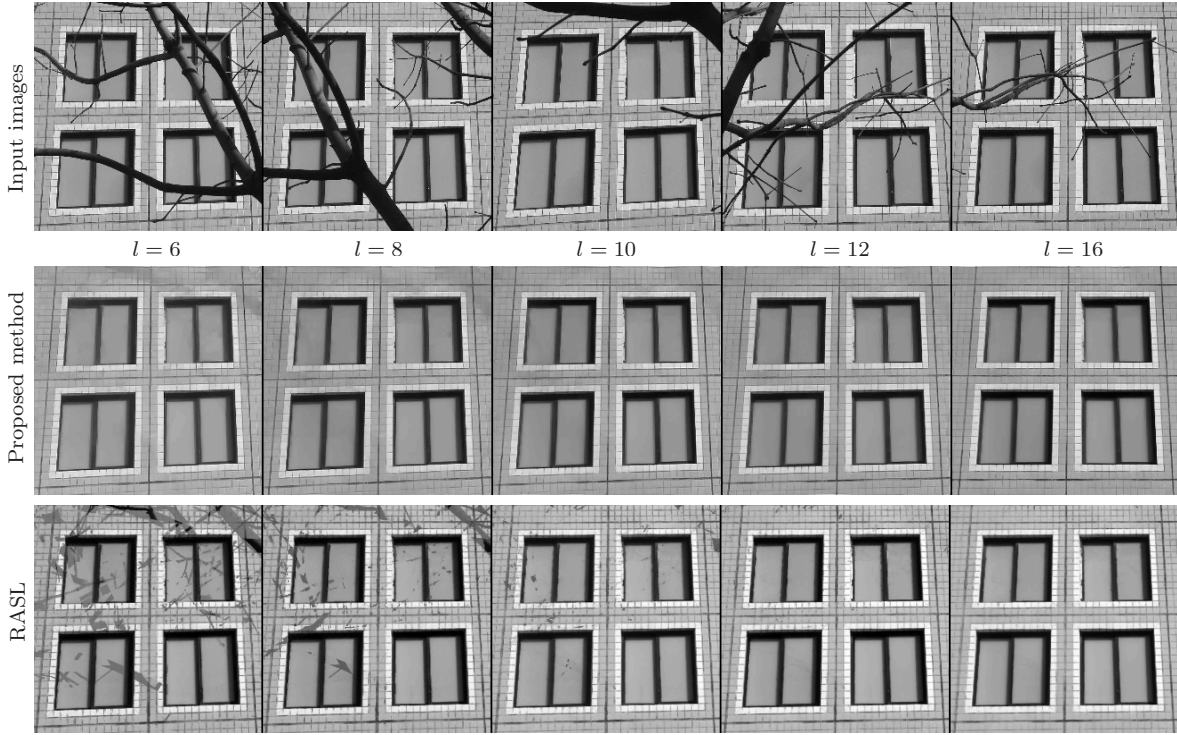
Figure 4.1. *Top panel: first 5 input images. Bottom panels: estimated background images with the proposed method and RASL using l = 6, 8, 10, 12 or 16 input images (from left to right).*

model mismatch by considering as "occlusions" the parts of the scene that cannot be aligned with the assumed model. Finally, let us also highlight that the requirements of Theorem 3.1 hold for a large class of transformation models $\tau_{\boldsymbol{\theta}}$ and leave us the possibility to choose more general type of transformations with more degrees of freedom than the one described above.

Finally, another limitation concerns the number of transformation parameters that the algorithm can efficiently handle. Indeed, if the number q of transformation parameters is large, the minimization problem at the second step of Algorithm 1 might become intractable. First, the computation of the matrix $\mathsf{H} \in \mathbb{R}^{q \times q}$, defined in (2.6), will become time consuming (unless the transformation model used exhibits special structure). Second, solving the minimization problem (2.7) is computationally expensive for unstructured large matrix H. Note however that one way to circumvent this issue is to approximate the matrix H by computing only its most "significant" entries and setting the remaining ones to zero.

4. Experiments. In this section, we demonstrate numerically the efficiency of the algorithm in three different applications: robust image alignment, compressed sensing, and super-resolution. The code and data needed to reproduce the results presented in this section are available at `lts2www.epfl.ch/people/gilles/softwares`.

4.1. Robust image alignment. Robust image alignment consists in aligning several images of a given scene despite the presence of large occlusions, as, for example, in the images

$l = 8$ $l = 16$

Proposed method
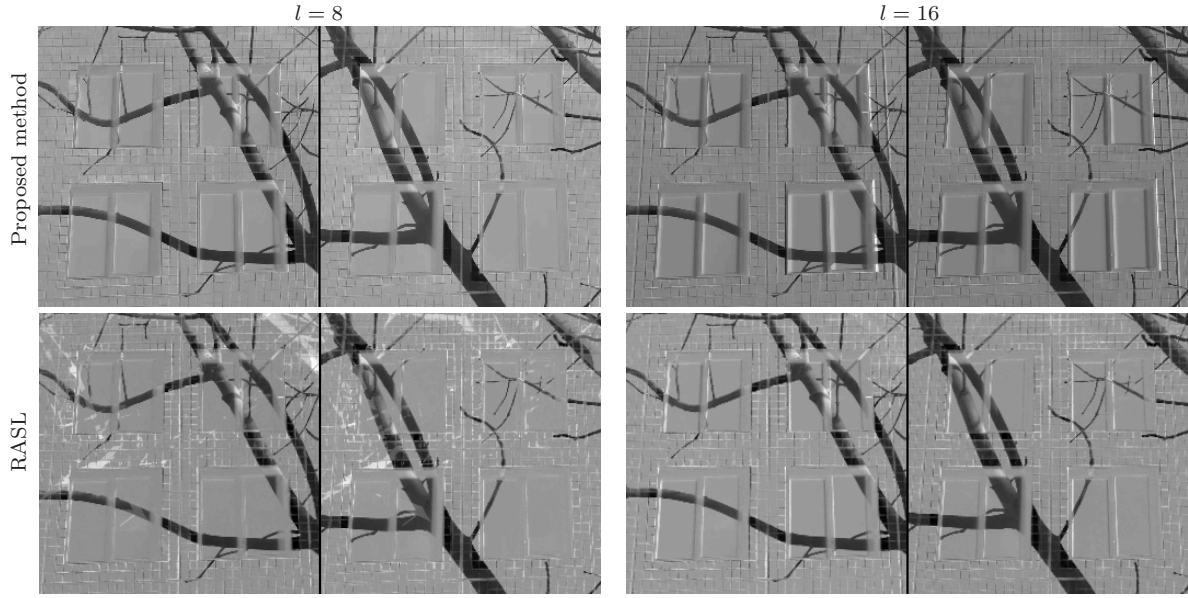
RASL

Figure 4.2. *First* 2 *estimated foreground images with the proposed method and RASL using* 8 *(left) or* 16 *(right) input images.*

Not registered Registered - Proposed method Registered - RASL
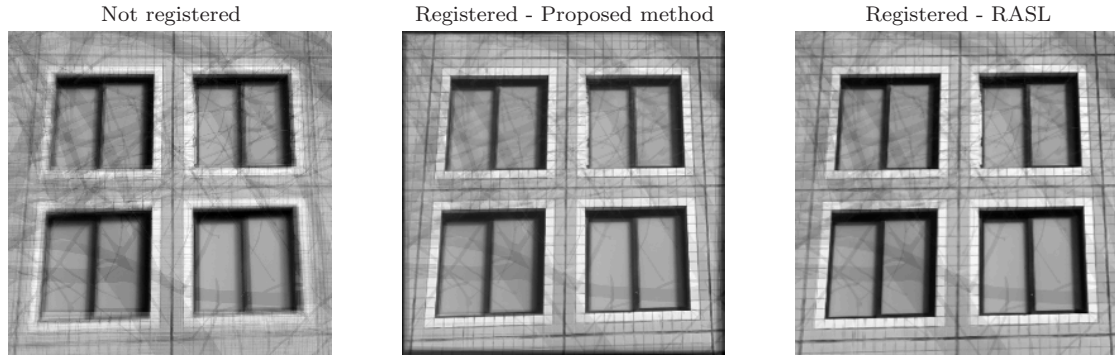
Figure 4.3. *Left panel: superposition of the first* 10 *input images. Middle panel: superposition of the same* 10 *images after registration with our method. Right panel: superposition of the same* 10 *images after registration with RASL.*

x_0^1 x_0^{11} x_0^{21} x_0^*
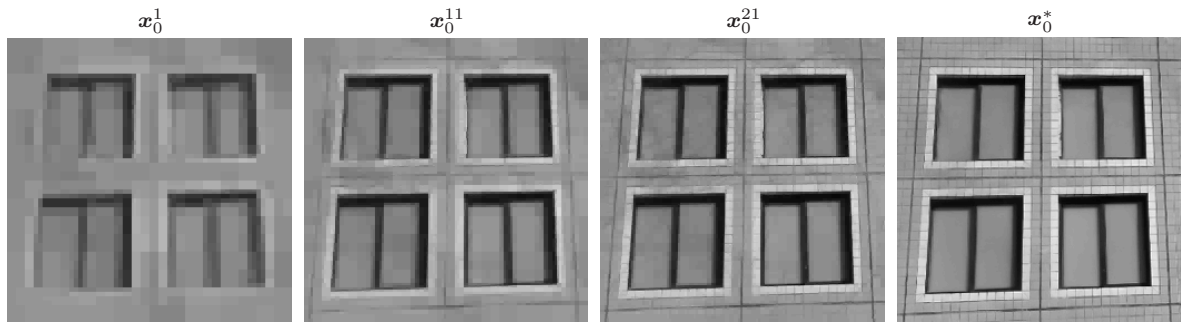
Figure 4.4. *Sequence of estimated background images* x_0^{k+1} *at* $k = 0$ *(left),* $k = 10$ *(middle left),* $k = 20$ *(middle right), and at convergence (right).*

presented in the top panel[4] of Fig. 4.1.

We propose here to solve this problem by using measurement model (1.2) and Algorithm 1 to minimize (2.2). In the measurement model, the matrices $\mathsf{A}_1, \ldots, \mathsf{A}_l$, are set to the identity $\mathsf{I}_n \in \mathbb{R}^{n \times n}$, and $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_l$, contain the occluded observed images. Ideally our algorithm should estimate the background image $\boldsymbol{x}_0$, i.e., the wall and the windows in Fig. 4.1 (top panel), extract the occlusions $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_l$, i.e., the branches, and provide the transformation parameters $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_l$, which align the background image on the observed images.

To show the efficiency of our technique, we compare our results with the ones obtained using RASL [25, 26]. This method is based on low rank and sparse approximations. More precisely, the authors in [25, 26] concatenate the occluded input images in a matrix $\mathsf{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_l] \in \mathbb{R}^{m \times l}$ and propose to solve

$$(\tilde{\mathsf{B}}, \tilde{\mathsf{F}}, \tilde{\boldsymbol{\theta}}) \in \underset{\mathsf{B}, \mathsf{F}, \boldsymbol{\theta}}{\operatorname{argmin}} \|\mathsf{B}\|_* + \frac{1}{\sqrt{m}} \|\mathsf{F}\|_{1,1} \quad \text{subject to} \quad \mathsf{Y} \circ \tau_{\boldsymbol{\theta}} = \mathsf{B} + \mathsf{F}, \tag{4.1}$$

where $\mathsf{B} \in \mathbb{R}^{m \times l}$, $\mathsf{F} \in \mathbb{R}^{m \times l}$, $\|\mathsf{B}\|_*$ denotes the nuclear norm of B, $\|\mathsf{F}\|_{1,1}$ is the sum of the absolute value of the entries of F, and $\mathsf{Y} \circ \tau_{\boldsymbol{\theta}}$ are the input images transformed according to the parameters $\boldsymbol{\theta}$ and the transformation model $\tau_{\boldsymbol{\theta}}$. This optimization problem decomposes the initial images Y in a low-rank matrix B which contains the background elements, and a sparse matrix F which contains the occlusions. The transformation parameters $\boldsymbol{\theta}$ are adjusted so that the matrix F is as sparse as possible and the rank of the matrix B is as small as possible.

One can remark that the data constraint $\mathsf{Y} \circ \tau_{\boldsymbol{\theta}} = \mathsf{B} + \mathsf{F}$ is non-linear in $\boldsymbol{\theta}$ and minimizing (4.1) is thus not trivial. To get around this difficulty, the authors linearize the term $\mathsf{Y} \circ \tau_{\boldsymbol{\theta}}$ with respect to the transformation parameters. This approximation is only valid for small transformations but the optimization problem becomes convex and easier to solve. Then, to align images with large transformations between them, the authors propose to repeatedly solve this convex problem and linearize the data constraint around the new estimated parameters to improve, little by little, the registration. Note that the authors do not provide a rigorous proof of convergence of this procedure.

Finally, it is interesting to note that, surprisingly, no constraints are imposed on the transformation parameters in (4.1). Therefore, if there exist a set of parameters $\boldsymbol{\theta}_0 \in \mathbb{R}^{lq}$ such that $\mathsf{Y} \circ \tau_{\boldsymbol{\theta}_0} = \mathsf{0}$, then it is easy to prove that the global minimum of (4.1) is 0 and $\tilde{\mathsf{B}} = \tilde{\mathsf{F}} = \mathsf{0}$. In such a situation, finding a global minimizer of (4.1) obviously does not yield to the desired solution, and, if one correctly aligns the images, it is only because the procedure described in [25, 26] converges to a local minima of problem (4.1). Note that as soon as the class of transformations considered allows us to scale the images, we already have $\mathsf{Y} \circ \tau_{\boldsymbol{\theta}} \to \mathsf{0}$ when the scaling parameters tends to zero. Let us however highlight that this issue can trivially be corrected by adding a constraint on the transformation parameters in the original formulation (4.1), and slightly modify the procedure described in (4.1) to correct this minor problem.

4.1.1. Background/foreground separation quality. In this first set of experiments, we use $l = 6, 8, 10, 12$, and 16 occluded images of $n = 256 \times 256$ pixels as inputs. The first 5 images are presented in the top panels of Fig. 4.1. Then, for each l, we run Algorithm 1 and RASL to align the images and to estimate the background and foreground images.

---

[4]Dataset available at `perception.csl.illinois.edu/matrix-rank/rasl.html`.

In our algorithm, we set $\kappa = 100$, $\lambda_{\boldsymbol{\theta}} = 0.1$, $\lambda_{\boldsymbol{x}}^k = \max(0.9^k\,(20\kappa), 0.1)$, and $\mu = 10^{-10}$. The matrix $\boldsymbol{\Psi}$ is build using the Haar wavelet basis, and $f(\boldsymbol{x})$ is set to $\|\boldsymbol{\Psi}^{\mathsf{T}}\boldsymbol{x}\|_1$. We also assume that the transformations between images are well modeled by homographies and thus use model (3.5) for $\tau_{\boldsymbol{\theta}}$. The parameters are constrained as follows: $|1 - \theta_1|, |1 - \theta_5| \leqslant 0.2$, $|\theta_2|, |\theta_4| \leqslant 0.2$, $|\theta_3|, |\theta_6| \leqslant 20$, $|\theta_7|, |\theta_8| \leqslant 0.01$, with u_1 and u_2 belonging to $\{-127, -126, \ldots, 128\}$.

In Fig. 4.1, we present the estimated background images[5] with our method and RASL from different number of occluded input images. One can first remark that with a small number of input images, i.e., $l = 6, 8$ and 10, the background images estimated with our method contains much less artifacts due to the occlusions than the ones estimated with RASL. Finally, our method only needs 10 input images to obtain a background image free of occlusion artifacts while RASL needs more input images.

In Fig. 4.2, we present foreground images estimated with both methods from $l = 8$ and 16 input images. One can notice that, for both methods, the foreground images essentially contain the occlusions, i.e., the branches, present in the initial input images. However, at $l = 8$, we still remark that the foreground images estimated with RASL are contaminated by objects not initially present in the input images. For a small number of input images, our method separates the background from the foreground better than RASL. At $l = 16$, the foreground images estimated by both method are quite similar. We note however that the windows are slightly less visible in the foreground images estimated by RASL.

To visualize the quality of the estimated transformation parameters, we present in Fig. 4.3 the superposition of $l = 10$ input images before and after registration with our method and RASL. One can easily remark that both methods are able to accurately register the set of input images.

Finally, we present in Fig. 4.4 the evolution of estimated background image $\boldsymbol{x}_0^{k+1}$ at iterations $k = 0$, $k = 10$, $k = 20$, $k = 30$, and at convergence with our method. This sequence of images illustrates the fact that, as explained in Section 2.2, the images are reconstructed in a coarse-to-fine scales manner with our method.

4.1.2. Registration quality. We now compare quantitatively the registration quality that one can obtain with RASL and our method.

In this second set of experiments, we use a "scaling+translational" model represented by 3 parameters $\boldsymbol{\theta} = (s, t_x, t_y)^{\mathsf{T}} \in \mathbb{R}^3$ with

$$\tau_{\boldsymbol{\theta}}(\boldsymbol{u}) = (s\,u_1 + t_x, \; s\,u_2 + t_y)^{\mathsf{T}}, \tag{4.2}$$

where s is a scaling factor, and (t_x, t_y) are translation parameters in the first and second dimensions of the images, respectively. Then, to compare the registration performance, we run two different types of simulations on images of $n = 128 \times 128$ pixels. The spatial coordinates u_1 and u_2 thus belong to $\{-63, -62, \ldots, 64\}$. Let us mention that with this transformation model, all the requirements needed to apply Theorem 3.1 are met.

For the first type of simulations, we draw independently and uniformly at random l translation parameters, $\{(t_x)_1, \ldots, (t_x)_l\}$ and $\{(t_y)_1, \ldots, (t_y)_l\}$, in the intervals $[-\Delta t_x/2, \, \Delta t_x/2]$

---

[5]For RASL, the background image is obtained by averaging the estimated background images in the matrix $\tilde{\mathsf{B}}$.
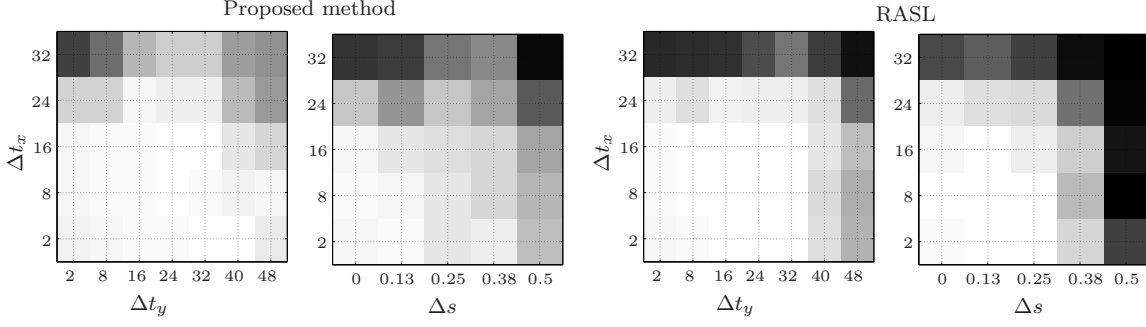
Figure 4.5. *Registration error σ averaged over 30 simulations for the proposed method (left panels) and RASL (right panels) as a function of $(\Delta t_x, \Delta t_y)$ and of $(\Delta t_x, \Delta s)$. The colorbar goes from white for $\sigma = 0$ to black for $\sigma = 0.5$.*

and $[-\Delta t_y/2, \Delta t_y/2]$, respectively, while keeping s fixed at 1. Then, we transform a reference image[6] according to these parameters to create l different input images. Finally, we register these images using RASL and our algorithm. This procedure is repeated for different sizes of the intervals, $\Delta t_x \in \{2, 8, 16, 24, 32\}$ and $\Delta t_y \in \{2, 8, 16, 24, 32, 40, 48\}$, and 30 independent simulations are performed for each pair of intervals.

For the second type of simulations, we draw independently and uniformly at random l translation parameters $\{(t_x)_1, \ldots, (t_x)_l\}$ in the intervals $[-\Delta t_x/2, \Delta t_x/2]$ and l scaling parameters $\{s_1, \ldots, s_l\}$ in the interval $[1 - \Delta s/2, 1 + \Delta s/2]$, while keeping t_y fixed at 0. This procedure is repeated with $\Delta t_x \in \{2, 8, 16, 24, 32\}$ and $\Delta s \in \{0, 0.13, 0.25, 0.38, 0.5\}$, and 30 independent simulations are also performed for each pair of intervals.

In our algorithm, we set $\kappa = 100$, $\lambda_{\boldsymbol{\theta}} = 0.1$, $\lambda_{\boldsymbol{x}}^k = \max(0.9^k (50\kappa), 0.1)$, and $\mu = 10^{-10}$. The matrix $\boldsymbol{\Psi}$ is build using the Daubechies 8 wavelet basis, and $f(\boldsymbol{x})$ is set to $\|\boldsymbol{\Psi}^{\mathsf{T}}\boldsymbol{x}\|_1$. The parameters are constrained as follows: $|1 - s| \leqslant 0.5$, and $|tx|, |ty| \leqslant 64$.

To compare the registration quality of both methods, we compute the normalized root-mean-square-error $\sigma_{i \to j}$ between the ground truth parameters $\boldsymbol{\theta}_{i \to j} \in \mathbb{R}^3$ that register the ith image onto the jth one and the corresponding parameters $\boldsymbol{\theta}_{i \to j}^* \in \mathbb{R}^3$ estimated by either RASL or our method. These values are then averaged over all possible combinations of images to obtain the registration error

$$\sigma = \frac{1}{l(l-1)} \sum_{i=1}^{l} \sum_{j \neq i} \sigma_{i \to j} \quad \text{with} \quad \sigma_{i \to j} = \frac{\|\boldsymbol{\theta}_{i \to j}^* - \boldsymbol{\theta}_{i \to j}\|_2}{\|\boldsymbol{\theta}_{i \to j}\|_2}.$$

Fig. 4.5 presents the registration error σ averaged over the 30 simulations obtained with RASL and our method for each type of simulations with $l = 10$. One can remark that for small transformations RASL is slightly more precise than our method. However, for large transformations, we obtain smaller registration error than RASL. Our method is thus more robust to large displacements. Note that we also performed these simulations with $l = 5$ and the same conclusions hold.

---

[6]The image used is available at `cvlab.epfl.ch/~strecha/multiview/rawMVS.html` and is part of the castle-R20 dataset [38].
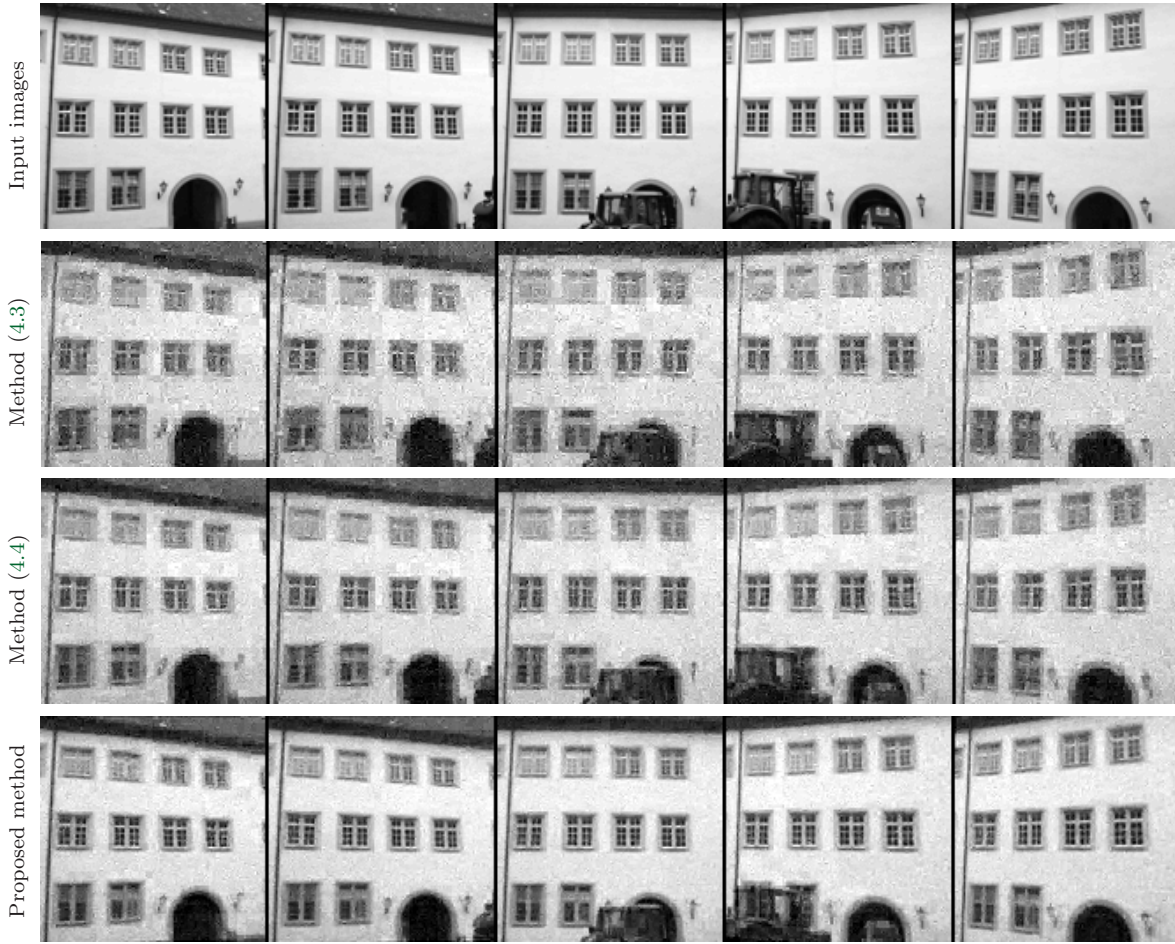
Figure 4.6. *Top panel: 5 initial images. Middle-top panel: reconstructed images with method (4.3). Middle-bottom panel: reconstructed images with method (4.4). Bottom panel: reconstructed images with the proposed method.*

4.2. Compressed sensing. In this second experiment, we study the performance of our algorithm in a compressed sensing setting, and compare it with other common algorithms.

We use $l = 5$ different images[7] of the same scene to generate 5 different measurement vectors $\boldsymbol{y}_1, \dots, \boldsymbol{y}_l$. These images, presented in the top panels of Fig. 4.6, contain $n = 128 \times 128$ pixels. Note that parts of the scene are sometimes occluded. The measurement vectors are obtained using the spread spectrum technique of [31], which consists of pre-modulating each image with a random ± 1 sequence before randomly selecting m Fourier coefficients. Let $\boldsymbol{c} = (c_1, \dots, c_n)^\mathsf{T} \in \mathbb{R}^n$ be a Rademacher sequence, $\mathsf{F} \in \mathbb{C}^{n \times n}$ be the matrix that implements the 2D discrete Fourier transform, and $\Omega_j \in \mathbb{N}^m$ be l sets of m indices selected independently and uniformly at random in $\{1, \dots, n\}$. As we are dealing with images living in $\mathbb{R}^n$, we can

---

[7]All images are in the castle-R20 dataset available at `cvlab.epfl.ch/~strecha/multiview/rawMVS.html` [38].
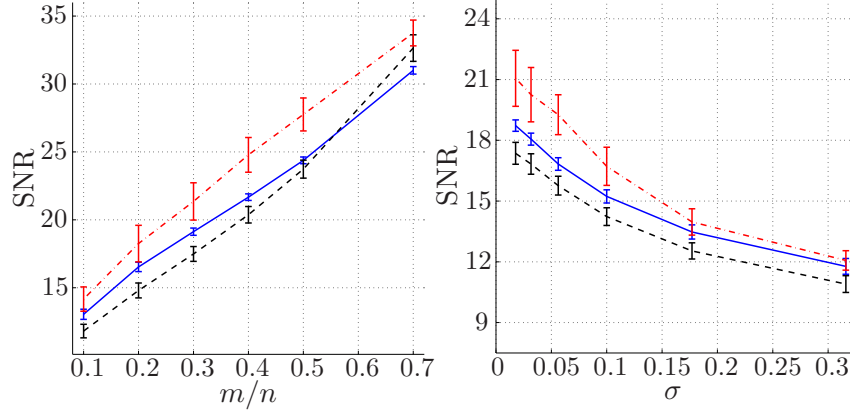
Figure 4.7. *Left panel: Reconstruction SNR as a function of m/n. Right panel: Reconstruction SNR as a function of the noise level σ. All curves represents the mean SNR over the 30×5 reconstructed images. The vertical lines shows the variation of the SNR at one standard deviation. The dashed black, continuous blue, and dot-dashed red curves represent the reconstruction SNR for method (4.3), (4.4), and our algorithm, respectively.*

Figure 4.8. *Left panel: superposition of the 5 initial images without registration. Middle panel: superposition of the 5 initial images after registration with the estimated parameters $\boldsymbol{\theta}_1^*, \ldots, \boldsymbol{\theta}_5^*$. Right panel: estimated background image $\boldsymbol{x}_0^*$.*

restrict the selection of the Fourier coefficients to one half of the Fourier plane. We denote $\tilde{\mathsf{F}} \in \mathbb{C}^{n/2 \times n}$ the restriction of F to this half plane. The observation matrices satisfy

$$\mathsf{A}_j = \begin{pmatrix} \tilde{\mathsf{F}}^{\mathrm{r}} \\ \tilde{\mathsf{F}}^{\mathrm{i}} \end{pmatrix}_{\Omega_j} \mathsf{C} \in \mathbb{R}^{m \times n}, \quad j = 1, \ldots, l,$$

where $\tilde{\mathsf{F}}^{\mathrm{r}}, \tilde{\mathsf{F}}^{\mathrm{i}} \in \mathbb{R}^{n/2 \times n}$ are respectively the real and imaginary part of $\tilde{\mathsf{F}}$, $\mathsf{C} \in \mathbb{R}^{n \times n}$ is the diagonal matrix containing the sequence $\boldsymbol{c}$, and $(\cdot)_{\Omega_j}$ restricts an $n \times n$ matrix to its rows indexed by Ω_j.

We assume here that the transformations between images can be represented by homographies and use model (3.5) for $\tau_{\boldsymbol{\theta}}$. For the prior term f, we assume that the images are sparse in the Haar wavelet basis $\mathsf{W} \in \mathbb{R}^{n \times n}$. We build the block-diagonal matrix $\boldsymbol{\Psi}$ by repeating $l + 1$ times the matrix W on the diagonal, and set $f(\boldsymbol{x}) = \|\boldsymbol{\Psi}^{\mathsf{T}} \boldsymbol{x}\|_1$. The same matrix $\boldsymbol{\Psi}$ is used in the cost-to-move term h_μ. In all simulations, we use $\lambda_{\boldsymbol{\theta}} = 0.1$,

$\lambda_{\boldsymbol{x}}^k = \max(0.9^k (20\kappa), 0.1)$, $\mu = 10^{-10}$, and the following constraints apply on the transformation parameters: $|1 - \theta_1|, |1 - \theta_5| \leqslant 0.2$, $|\theta_2|, |\theta_4| \leqslant 0.2$, $|\theta_3|, |\theta_6| \leqslant 20$, $|\theta_7|, |\theta_8| \leqslant 0.01$, with u_1 and u_2 belonging to $\{-63, -62, \ldots, 64\}$.

The first set of simulations study the performance of Algorithm 1 in the absence of noise for different number of measurements: $m \in \{0.1n, 0.2n, 0.3n, 0.4n, 0.5n, 0.7n\}$. The second set of simulations study the performance of Algorithm 1 at $m = 0.3n$ for different noise levels. The noise vector $\boldsymbol{n}$ follows an i.i.d zero-mean Gaussian distribution of standard deviation $\sigma \in \{0.01, 0.0177, 0.0316, 0.0562, 0.1, 0.177, 0.316\}$. The squared ℓ_2-norm of the noise vector $\boldsymbol{n}/\sigma$ follows a chi-square distribution with lm degrees of freedom and we compute the bound on the noise level $\|\boldsymbol{n}\|_2^2 \leqslant \epsilon^2$ using the 99th percentile of this distribution. Then, κ is set to 100 in the noiseless case and is chosen so that $\|\mathsf{A}(\boldsymbol{\theta}^*)\boldsymbol{x}^* - \boldsymbol{y}\|_2 \approx \epsilon$ in the presence of noise[8].

For comparison, we also reconstruct the images by solving the following convex optimization problems:

$$\min_{\mathsf{X}} \|\mathsf{W}^\mathsf{T}\mathsf{X}\|_1 \quad \text{subject to} \quad \|\mathsf{Y} - \mathsf{A}\mathsf{X}\|_F \leqslant \epsilon, \tag{4.3}$$

and

$$\min_{\mathsf{X}} \|\mathsf{W}^\mathsf{T}\mathsf{X}\|_{2,1} \quad \text{subject to} \quad \|\mathsf{Y} - \mathsf{A}\mathsf{X}\|_F \leqslant \epsilon, \tag{4.4}$$

where $\mathsf{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_l) \in \mathbb{R}^{n \times l}$, $\mathsf{Y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_l) \in \mathbb{R}^{m \times l}$, $\|\cdot\|_F$ is the Frobenius norm, $\|\mathsf{X}\|_1 = \sum_{i=1}^n \sum_{j=1}^l |x_{ij}|$, and $\|\mathsf{X}\|_{2,1} = \sum_{i=1}^n (\sum_{j=1}^l x_{ij}^2)^{1/2}$, with x_{ij} the entry on the ith line and jth column of X. In the first problem, the images $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_l$ are reconstructed with a sparse prior in the Haar wavelet basis. This is an extension of the Basis Pursuit problem, advocated by the compressed sensing theory [12], to multiple signals. In the second problem, the prior term favors reconstructions which have a similar sparsity pattern in the Haar wavelet basis [5], therefore imposing some correlation between images.

Fig. 4.7 shows the quality of the reconstructed images in term of SNR. Let $(\boldsymbol{x}^*, \boldsymbol{\theta}^*) \in \mathbb{R}^{(l+1)n} \times \mathbb{R}^{ln}$ be the images and transformation parameters recovered with our algorithm and $\boldsymbol{x} \in \mathbb{R}^{ln}$ be the initial l images. The reconstruction SNR satisfies $-20 \log_{10}(\|\mathsf{S}(\boldsymbol{\theta}_j^*)\boldsymbol{x}_0^* + \boldsymbol{x}_j^* - \boldsymbol{x}_j\|_2/\|\boldsymbol{x}_j\|_2)$. For the two other methods (4.3) and (4.4), the reconstruction SNR satisfies $-20 \log_{10}(\|\boldsymbol{x}_j^* - \boldsymbol{x}_j\|_2/\|\boldsymbol{x}_j\|_2)$ where $\mathsf{X}^* = (\boldsymbol{x}_1^*, \ldots, \boldsymbol{x}_l^*) \in \mathbb{R}^{n \times l}$ are the reconstructed images with these methods. The first graph of Fig. 4.7 shows the reconstructions SNR as a function of m/n in the absence of noise, and the second graph shows the reconstructions SNR as a function of the noise level σ for $m/n = 0.3$. In each case, we perform 30 simulations with independent noise realization and choice of Ω_j. The curves show the mean SNR over the 30×5 reconstructed images. The vertical lines indicate the variation of the SNR at one standard deviation. One can deduce from these curves that the best reconstructions are obtained with our method.

In Fig. 4.6, we present the ground truth images as well as reconstructions obtained with the three different methods for $m/n = 0.3$ in the absence of noise. On can remark that the reconstructions obtained with our algorithm exhibits much finer details. Fig. 4.8 shows the estimated background image, next to the superposed initial images before and after registration with the estimated parameters. One can also note that the estimated background image

---

[8]For each noise level, we use one simulation to estimate a value of κ such that, at converge of the algorithm, $\|\mathsf{A}(\boldsymbol{\theta}^*)\boldsymbol{x}^* - \boldsymbol{y}\|_2 \in [0.99\epsilon, 1.01\epsilon]$. We then use the same value of κ for all subsequent simulations.
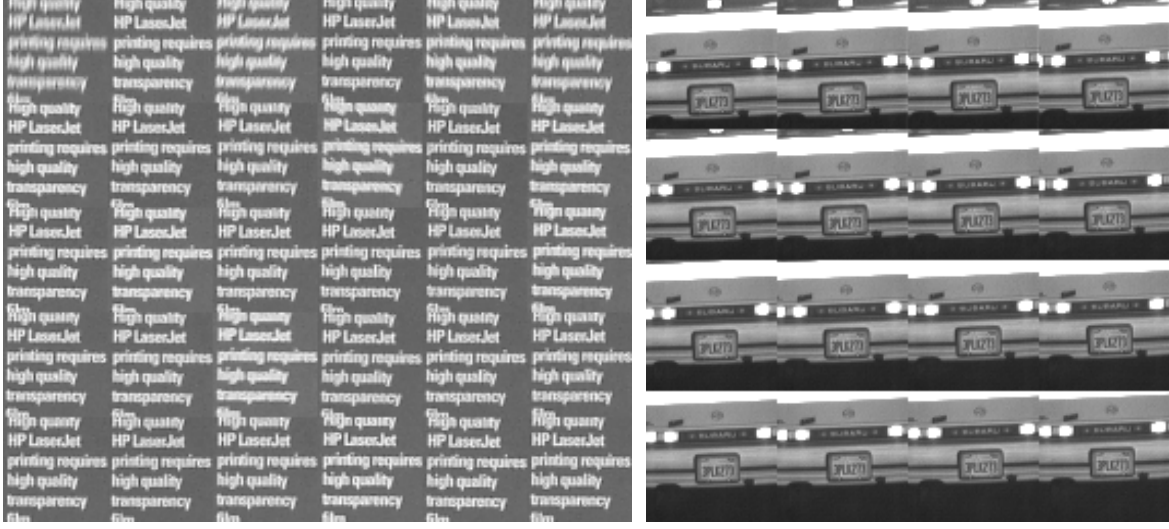
Figure 4.9. *Multi-frame datasets used to test the efficiency of our algorithm for super-resolution. The dataset on the right contains* 30 *low-resolution images of* $m = 48 \times 48$ *pixels. The dataset on the left contains* 16 *low-resolution images of* $m = 64 \times 64$ *pixels.*

is free of occlusions, and that the initial images are better aligned after registration with the estimated parameters.

4.3. Super-resolution.

In this last experiment, we demonstrate the performance of our algorithm in super-resolution from multiple frames and compare it with the method described in [44], whose code is available at `lcav.epfl.ch/software/superresolution`. We test both algorithms on the datasets[9] presented in Fig. 4.9.

To increase the resolution of the available images by a factor of 2 in both dimensions with our method, we use the measurement model (1.2) where the sensing matrices A_j, with $j = 1, \ldots, l$, model a blurring and downsampling operator that averages all the pixels in a rectangular window of 2×2 pixels and uniformly downsamples the resulting blurred image by a factor of 2 in both dimensions. The sensing matrices are identical for all j. The vectors $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_l$ are initialized with the low-resolution images. The transformations between images are assumed to be affine, and we use model (3.4) for $\tau_{\boldsymbol{\theta}}$. The transformation parameters are constrained as follows: $|1 - \theta_1|, |1 - \theta_5| \leqslant 0.2$, $|\theta_2|, |\theta_4| \leqslant 0.2$, $|\theta_3|, |\theta_6| \leqslant 30$, with u_1 and u_2 belonging to $\{-n/2 + 1, \ldots, n/2\}$. The matrix $\boldsymbol{\Psi}$ is build using the Haar wavelet basis. We also assume that the high-resolution images $\boldsymbol{x}_0, \ldots, \boldsymbol{x}_l$ are sparse in the gradient domain and set $f(\boldsymbol{x}) = \sum_{j=0}^{l} g(\boldsymbol{x}_j)$ in (2.2), where $g \colon \mathbb{R}^n \to \mathbb{R}$ is the isotropic Total Variation (TV) norm. Let us recall that the vector $\boldsymbol{x}_j$ contains the samples $x_j(\boldsymbol{u}_k)$, with $k = 1, \ldots, n$ and

---

[9]Datasets available at `users.soe.ucsc.edu/~milanfar/software/sr-datasets.html` (credit: Peyman Milanfar)

Bicubic interp.	[44] + Bicubic interp.	[44] + [27]	Proposed method
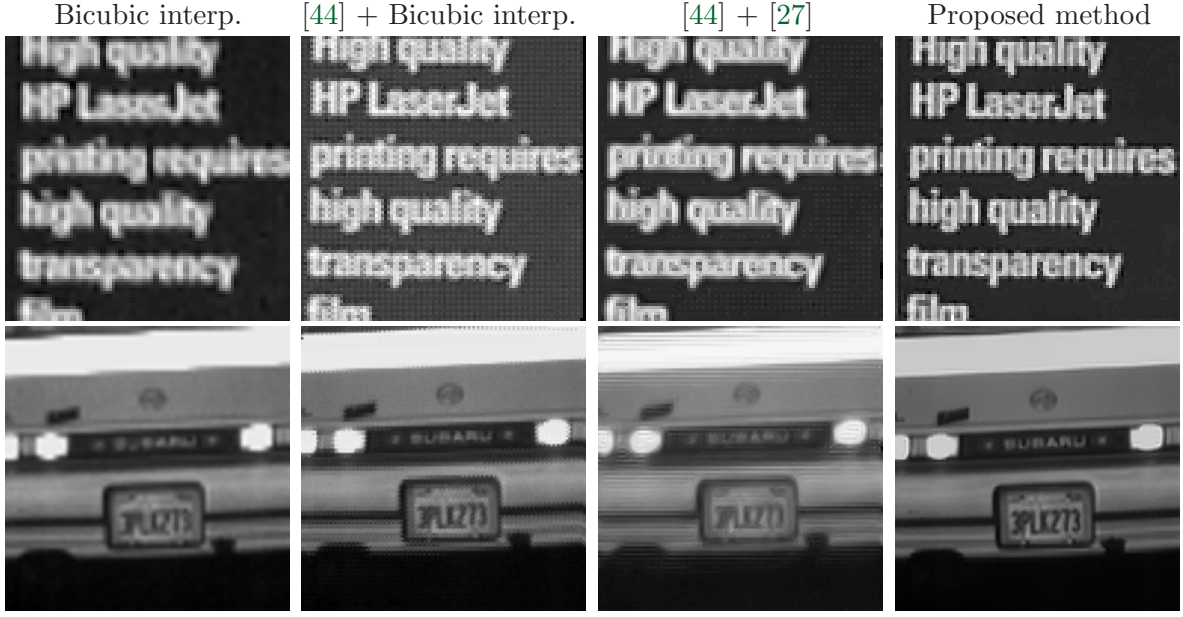
Figure 4.10. *From left to right: one input image of the original dataset upsampled independently using a bicubic interpolation; super-resolved image using the method described in [44] in combination with a bicubic interpolation; super-resolved image using the method described in [44] in combination with [27]; super-resolved background image estimated by our algorithm.*

$\boldsymbol{u}_k = (u_1^k, u_2^k)$ (see Section 1.1). The isotropic TV norm g satisfies

$$g(\boldsymbol{x}_j) = \sum_{k=1}^{n} \sqrt{|\nabla_1 \, x_j(\boldsymbol{u}_k)|^2 + |\nabla_2 \, x_j(\boldsymbol{u}_k)|^2}, \text{ where } \begin{cases} \nabla_1 \, x_j(\boldsymbol{u}_k) = x_j(u_1^{k+1}, u_2^k) - x_j(u_1^k, u_2^k), \\ \nabla_2 \, x_j(\boldsymbol{u}_k) = x_j(u_1^k, u_2^{k+1}) - x_j(u_1^k, u_2^k), \end{cases}$$

assuming that $x(u_1^{n+1}, \cdot) = x(\cdot, u_2^{n+1}) = 0$. Note that g and, consequently, f are semi-algebraic. Finally, we run Algorithm 1 to minimize (2.2) with $\kappa = 100$, $\lambda_{\boldsymbol{\theta}} = 0.1$, $\lambda_{\boldsymbol{x}}^k = \max(0.9^k \, (20\kappa), 0.1)$, and $\mu = 10^{-10}$.

The first dataset contains $l = 30$ low-resolution images of $m = 48 \times 48$ pixels. The super-resolved images thus contain $n = 96 \times 96$ pixels. In the top panel of Fig. 4.10, we present one super-resolved image obtained by upsampling independently one of the original images using a bicubic spline interpolation, our super-resolved background image, and two super-resolved images obtained with the code of [44]. These last two images have been obtained by first registering the low resolution images with the technique described in [44]. The first high-resolution image has then been obtained by using a bicubic interpolation to combine the information of all registered low-resolution images. The second high-resolution image has been obtained using the technique described in [27]. On this first dataset, one can easily remark that our super-resolved image exhibits much shaper details than the images obtained with the other methods. In particular, the contours of the letters are better defined and the text is easier to read.

The second dataset contains $l = 16$ low-resolution images of $m = 64 \times 64$ pixels. The super-resolved images thus contain $n = 128 \times 128$ pixels. In Fig. 4.10, we present one super-resolved

Figure 4.11. *Super-resolved background image estimated by our algorithm using Haar wavelets (left) and Daubechies wavelets (right) for* Ψ *in the cost-to-move function* h_μ.

image obtained by upsampling independently one of the original images using a bicubic spline interpolation, our super-resolved background image, and two super-resolved image obtained with the code of [44], as described before. On this second dataset, one can notice that the brand of the car and license plate number are much better reconstructed with our method. The details are sharper, confirming again the superiority of our technique.

Let us mention the fact that the registration quality obtained with the method described in [44] is very sensitive to the choice of the reference frame. The results presented here are the best we were able to obtain after a careful choice of the reference frame. For most of the other choices of reference frame, this method was not able to register properly the low-resolution images and the subsequent super-resolved images were suffering of strong artifacts. Note that our method does not suffer of this issue as the algorithm automatically selects a reference coordinate system within the allowed range Θ.

Finally, to show the effect of the choice of Ψ in the cost-to-move function h_μ, we run again Algorithm 1 by building Ψ using Daubechies 8 wavelets instead of the Haar wavelets. We present in Fig. 4.11 the reconstructions obtained with both wavelet bases. We do not observe any significant differences between both reconstructions. The choice of Ψ in the cost-to-move term h_μ thus does not seem to affect significantly the quality of the final reconstruction.

4.4. Running time of the algorithm. We finally conclude this section by mentioning the computation times required to minimize the non-convex functional (2.1) using our Matlab implementation of Algorithm 1. The running times mentioned have been obtained using a desktop computer (Windows 7, Intel Core i7 3.50Ghz, 16GB of RAM) and the parallel toolbox of Matlab to estimate the transformation parameters simultaneously.

For the robust image alignment problem, our implementation of the algorithm has been running for about 14 minutes to align 6 images of 256×256 pixels. For the compressed sensing problem, around 7 minutes have been necessary to reconstruct 5 images of 128×128 pixels from 30 percent of measurements. Finally, for the super-resolution problem, it has taken about 21 minutes to estimate a high resolution image of the car with 128×128 pixels from the 16 low-resolution input images.

Compared to the other methods we tested, the running time of our algorithm is longer.

However, we would like to highlight that our implementation is general to be able to handle all these examples. We have not used the specificity of each problem to optimize the code and reduce the computation time. Furthermore, we have noticed that only a few iterations of the algorithm are necessary to estimate the transformation parameters. The last iterations rather serve the estimation of the background and foreground images. Therefore, an approximate solution can be obtained by stopping the algorithm after a few iterations and then solve one last time problem (2.3) with $\lambda_{\boldsymbol{x}} = 0$ to estimate the images. Note however that the sequence of estimates does not converge to critical point of L with such an approach.

5. Conclusion. We presented a method to solve multi-view imaging problems where one has to reconstruct several images of a scene from only a few linear observations made at different viewpoints. Each observed image is modeled as the sum of a geometrically transformed background image and of a foreground image, modeling possible occlusions. We considered here global geometric transformations represented by few parameters, such as translations, affine transformations, or homographies. The proposed reconstruction method jointly estimates the images (background and foreground) and the transformation parameters by minimizing a non-convex functional. We studied the convergence of the proposed algorithm and showed that the generated sequence of estimates converges to a critical point of the non-convex functional for commonly used priors and transformation models. Numerical experiments show that the proposed algorithm is competitive with well-known methods in robust image alignment, compressed sensing, and super-resolution. The method is able to robustly register a set of images in the presence of large displacements, and the separation background/foreground is also very accurate, even when only few random measurements of the images are accessible.

Finally, we would like to conclude by highlighting the potential interest of the proposed technique in free breathing coronary magnetic resonance imaging (MRI) [37]. One of the major challenges in this application is to handle respiratory motion properly. The low acquisition speed in MRI forces researchers to invent new techniques to compensate for the inevitable motions occurring during the acquisition. To avoid motions due to heart contractions, an ECG signal is acquired to ensure that the Fourier measurements are always taken at the same instant in the heart cycle. A few measurements are then taken at each cycle. As the number of measurements acquired during one cycle is too small to accurately reconstruct an image of the heart, one should combine the measurements of several cycles to gather enough information. It is thus mandatory to properly compensate for the respiratory motion occurring between heart beats in the reconstruction method. Note that simple geometric transformations, such as translations, are already sufficient to reach good image quality [11]. Consequently, as the technique developed in this paper automatically compensates for such geometric transformations, it is a promising method for such an application. Note that this hope is confirmed by the preliminary results presented in [30] on real MRI data.

Appendix A. To establish the third point of Theorem 3.1, we need the following theorem establish by Attouch *et al.* in [4].

Theorem A.1 (Theorem 6.2, [4]). *Let p be a positive integer larger than 2, $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_p)$ be a vector in $\mathcal{H} = \mathbb{R}^{n_1} \times \ldots \times \mathbb{R}^{n_p}$, and $g \colon \mathcal{H} \to \mathbb{R} \cup \{+\infty\}$ a function of the form $g(\boldsymbol{x}) = Q(\boldsymbol{x}) + \sum_{i=1}^{p} g_i(\boldsymbol{x}_i)$, where $Q \colon \mathcal{H} \to \mathbb{R}$ is a $\mathcal{C}^1$ function with locally Lipschitz continuous gradient and $g_i \colon \mathbb{R}^{n_i} \to \mathbb{R} \cup \{+\infty\}$ is a proper lower semicontinuous function, $i = 1, \ldots, p$.*

Assume that g is bounded from below and satisfies the Kurdyca-Łojasiewicz property. Let
$(\boldsymbol{x}^k)_{k\in\mathbb{N}}$ *and* $(\boldsymbol{\nu}^k)_{k\in\mathbb{N}}$ *be sequences such that*

$$
\begin{aligned}
g_i(\boldsymbol{x}_i^{k+1}) &+ Q(\boldsymbol{x}_1^{k+1},\ldots,\boldsymbol{x}_{i-1}^{k+1},\boldsymbol{x}_i^{k+1},\ldots,\boldsymbol{x}_p^k) + a\|\boldsymbol{x}_i^{k+1} - \boldsymbol{x}_i^k\|_2^2 \\
&\leqslant g_i(\boldsymbol{x}_i^k) + Q(\boldsymbol{x}_1^{k+1},\ldots,\boldsymbol{x}_{i-1}^{k+1},\boldsymbol{x}_i^k,\ldots,\boldsymbol{x}_p^k), \\
\boldsymbol{\nu}_i^{k+1} &\in \partial g_i(\boldsymbol{x}_i^{k+1}), \\
\|\boldsymbol{\nu}_i^{k+1} &+ \partial_{x_i} Q(\boldsymbol{x}_1^{k+1},\ldots,\boldsymbol{x}_i^{k+1},\boldsymbol{x}_{i+1}^k,\ldots,\boldsymbol{x}_p^k)\|_2 \leqslant b\|\boldsymbol{x}_i^{k+1} - \boldsymbol{x}_i^k\|_2,
\end{aligned}
\tag{A.1}
$$

where i ranges over $\{1,\ldots,p\}$ *and* $a,b > 0$*. Then, if the sequence* $(\boldsymbol{x}^k)_{k\in\mathbb{N}}$ *is bounded, it converges to some critical point* $\boldsymbol{x}^*$ *of g.*

The objective function L defined in (2.2) is equal to

$$
L(\boldsymbol{x},\boldsymbol{\theta}) = \kappa\, Q(\boldsymbol{x},\boldsymbol{\theta}) + f(\boldsymbol{x}) + i_\Theta(\boldsymbol{\theta}),
$$

where $Q\colon \mathbb{R}^{(l+1)n} \times \mathbb{R}^{lq} \to \mathbb{R}$ satisfies $Q(\boldsymbol{x},\boldsymbol{\theta}) = \|\mathsf{A}(\boldsymbol{\theta})\boldsymbol{x} - \boldsymbol{y}\|_2^2$. The objective function has the the form required by Theorem A.1. Furthermore, f and i_{Θ_i} are proper lower semicontinuous functions, and, using the assumptions of Theorem 3.1, Q is twice continuously differentiable and thus have locally Lipschitz continuous gradient. We also note that L is bounded below and satisfies the Kurdyca-Łojasiewicz property. To prove the third point of Theorem 3.1, it remains to show that the sequence generated by Algorithm 1 is bounded and satisfies conditions (A.1).

First, due to the constraints that apply on $\boldsymbol{\theta}$, it is obvious that the sequence $(\boldsymbol{\theta}^k)_{k\in\mathbb{N}}$ is bounded. Therefore, assuming that $(\boldsymbol{x}^k)_{k\in\mathbb{N}}$ is bounded, as in the third point of Theorem 3.1, is enough to ensure the boundedness condition of the entire sequence of estimates.

Second, $\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^k\|_2$ tends to 0 as $k \to +\infty$, as established in the second point of Theorem 3.1. Therefore there exists $k_0 \in \mathbb{N}$ such that for all $k \geqslant k_0$, $h_\mu(\Psi^\mathsf{T}(\boldsymbol{x}^{k+1} - \boldsymbol{x}^k)) = \|\Psi^\mathsf{T}(\boldsymbol{x}^{k+1} - \boldsymbol{x}^k)\|_2^2/(2\mu) = \|\boldsymbol{x}^{k+1} - \boldsymbol{x}^k\|_2^2/(2\mu)$, as $\Psi\Psi^\mathsf{T} = \mathsf{I}_{(l+1)n}$. Let k be larger than k_0 in the following. Inequality (2.4) shows that

$$
f(\boldsymbol{x}^{k+1}) + \kappa\, Q(\boldsymbol{x}^{k+1},\boldsymbol{\theta}^k) + \frac{\lambda}{4\mu}\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^k\|_2^2 \leqslant f(\boldsymbol{x}^k) + \kappa\, Q(\boldsymbol{x}^k,\boldsymbol{\theta}^k).
$$

Furthermore, the first order optimality condition of Problem (2.3) shows that there exists $\boldsymbol{\nu}_{\boldsymbol{x}}^{k+1} \in \partial f(\boldsymbol{x}^{k+1})$ such that

$$
\boldsymbol{\nu}_{\boldsymbol{x}}^{k+1} + \kappa\, \partial_{\boldsymbol{x}} Q(\boldsymbol{x}^{k+1},\boldsymbol{\theta}^k) + \frac{\lambda_{\boldsymbol{x}}^k}{2\mu}\,(\boldsymbol{x}^{k+1} - \boldsymbol{x}^k) = 0,
$$

which implies,

$$
\|\boldsymbol{\nu}_{\boldsymbol{x}}^{k+1} + \kappa\, \partial_{\boldsymbol{x}} Q(\boldsymbol{x}^{k+1},\boldsymbol{\theta}^k)\|_2 \leqslant \frac{\bar{\lambda}}{2\mu}\,\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^k\|_2.
$$

Finally, we deduce from equation (2.9) that

$$
i_\Theta(\boldsymbol{\theta}^{k+1}) + \kappa\, Q(\boldsymbol{x}^{k+1},\boldsymbol{\theta}^{k+1}) + \frac{\kappa\lambda_{\boldsymbol{\theta}}}{2}\|\boldsymbol{\theta}^{k+1} - \boldsymbol{\theta}^k\|_2^2 \leqslant i_\Theta(\boldsymbol{\theta}^k) + \kappa\, Q(\boldsymbol{x}^{k+1},\boldsymbol{\theta}^k).
$$

And, using the first order optimality condition of the problems (2.7), we conclude that there exists $\boldsymbol{\nu}_{\boldsymbol{\theta}_j}^{k+1} \in \partial i_{\Theta_j}(\boldsymbol{\theta}_j^{k+1})$ such that

$$\boldsymbol{\nu}_{\boldsymbol{\theta}_j}^{k+1} + \kappa \ \nabla Q_j^{k+1}(\boldsymbol{\theta}_j^k) + \kappa \ [\mathsf{H}(\boldsymbol{\theta}_j^k) + 2^i \lambda_{\boldsymbol{\theta}} \mathsf{I}_q](\boldsymbol{\theta}_j^{k+1} - \boldsymbol{\theta}_j^k) = 0,$$

for all $j = 1, \ldots, l$. Let Γ denote the Lipschitz constant of ∇Q on a product of balls $\mathcal{B}_{\boldsymbol{x}} \times \mathcal{B}_{\boldsymbol{\theta}}$ containing the sequence $(\boldsymbol{x}_k, \boldsymbol{\theta}_k)_{k \in \mathbb{N}}$. By construction of i in Algorithm 1, we necessarily have $2^i \lambda_{\boldsymbol{\theta}} \leqslant \eta = \max(2\Gamma, \lambda_{\boldsymbol{\theta}})$. Remark also that, on the same products of balls, their exists Λ such that the singular values of the matrices $\mathsf{H}(\boldsymbol{\theta}_j^k)$, $j = 1, \ldots, l$, are bounded above by Λ. Therefore,

$$\begin{aligned}
\|\boldsymbol{\nu}_{\boldsymbol{\theta}_j}^{k+1} + \kappa \ \nabla Q_j^{k+1}(\boldsymbol{\theta}_j^{k+1})\|_2 &= \|\boldsymbol{\nu}_{\boldsymbol{\theta}_j}^{k+1} + \kappa \nabla Q_j^k(\boldsymbol{\theta}_j^k) + \kappa \nabla Q_j^{k+1}(\boldsymbol{\theta}_j^{k+1}) - \kappa \nabla Q_j^{k+1}(\boldsymbol{\theta}_j^k)\|_2 \\
&\leqslant \|\boldsymbol{\nu}_{\boldsymbol{\theta}_j}^{k+1} + \kappa \nabla Q_j^{k+1}(\boldsymbol{\theta}_j^k)\|_2 + \kappa \|\nabla Q_j^{k+1}(\boldsymbol{\theta}_j^{k+1}) - \nabla Q_j^{k+1}(\boldsymbol{\theta}_j^k)\|_2 \\
&\leqslant \kappa(\Lambda + \eta) \ \|\boldsymbol{\theta}_j^{k+1} - \boldsymbol{\theta}_j^k\|_2 + \kappa\Gamma\|\boldsymbol{\theta}_j^{k+1} - \boldsymbol{\theta}_j^k\|_2 \\
&= \kappa(\Lambda + \eta + \Gamma) \ \|\boldsymbol{\theta}_j^{k+1} - \boldsymbol{\theta}_j^k\|_2,
\end{aligned}$$

which yields,

$$\|\boldsymbol{\nu}_{\boldsymbol{\theta}}^{k+1} + \kappa \ \partial_{\boldsymbol{\theta}} Q(\boldsymbol{x}^{k+1}, \boldsymbol{\theta}^{k+1})\|_2 \leqslant \kappa \ (\Lambda + \eta + \Gamma) \ \|\boldsymbol{\theta}^{k+1} - \boldsymbol{\theta}^k\|_2,$$

with $\boldsymbol{\nu}_{\boldsymbol{\theta}}^{k+1} = ((\boldsymbol{\nu}_{\boldsymbol{\theta}_1}^{k+1})^{\mathsf{T}}, \ldots, (\boldsymbol{\nu}_{\boldsymbol{\theta}_l}^{k+1})^{\mathsf{T}})^{\mathsf{T}} \in \partial i_{\Theta}(\boldsymbol{\theta}^{k+1})$. The sequence $(\boldsymbol{x}^k, \boldsymbol{\theta}^k)_{k \geqslant k_0}$ thus satisfies conditions (A.1) with $a = \min(\underline{\lambda}/(4\mu), \kappa\lambda_{\boldsymbol{\theta}}/2)$ and $b = \max(\bar{\lambda}/(2\mu), \kappa(\Lambda + \eta + \Gamma))$. This terminates the proof of the third point of Theorem 3.1.

REFERENCES

[1] H. Afshari, L. Jacques, L. Bagnato, A. Schmid, P. Vandergheynst, and Y. Leblebici, *The PANOPTIC camera: A plenoptic sensor with real-time omnidirectional capability.*, J. Signal Process. Syst. Signal Image Video Technol., accepted (2012).

[2] M. S. Asif, L. Hamilton, M. Brummer, and J. Romberg, *Motion-Adaptive Spatio-Temporal Regularization (MASTeR) for accelerated dynamic MRI*, Magn. Reson. Med., submitted, (2012).

[3] H. Attouch, J. Bolte, P. Redont, and A. Soubeyran, *Proximal alternating minimization and projection methods for nonconvex problems: an approach based on the Kurdyka-Lojasiewicz inequality*, Math. Oper. Res., 35 (2010), pp. 438–457.

[4] H. Attouch, J. Bolte, and B. F. Svaiter, *Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized Gauss–seidel methods*, Math. Program., (2011).

[5] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski, *Optimization with sparsity-inducing penalties*, Foundations and Trends in Machine Learning, 4 (2012), pp. 1–106.

[6] D. Baron, M. B. Wakin, M. F. Duarte, S. Sarvotham, and R. G. Baraniuk, *Distributed compressed sensing*, Tech. Report ECE-0612, Electrical and Computer Engineering Department, Rice University, 2006.

[7] H. H. Bauschke and P. L. Combettes, *Convex analysis and monotone operator theory in Hilbert spaces*, CMS Books in Mathematics, Springer New York, 2011.

[8] A. Beck and M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM J. Imaging Sci., 2 (2009), pp. 183–202.

[9] E. van den Berg, M. P. Friedlander, G. Hennenfent, F. Herrmann, R. Saab, and Ö. Yilmaz, *Sparco: A testing framework for sparse reconstruction*, Tech. Report TR-2007-20, Dept. Computer Science, University of British Columbia, Vancouver, October 2007.

[10] D. P. BERTSEKAS, *Projected Newton methods for optimization problems with simple constraints*, SIAM J. Control and Optimization, 20 (1982), pp. 221–246.

[11] G. BONANNO, G. PUY, Y. WIAUX, R. VAN HEESWIJK, AND M. STUBER, *About the performance of multi-dimensional radial self-navigation incorporating compressed sensing for free-breathing coronary MRI*, in International Society for Magnetic Resonance in Medicine (ISMRM) conference, Melbourne, 2012.

[12] E. J. CANDÈS, *Compressive sampling*, in Proc. International Congr. of Mathematicians, vol. 3, Madrid, Spain, 2006, pp. 1433–1452.

[13] X. CHEN AND P. FROSSARD, *Joint reconstruction of compressed multi-view images*, in Proc. IEEE Conf. Acoustics, Speech and Signal Processing, 2009, pp. 1005 – 1008.

[14] P. L. COMBETTES AND J.-C-PESQUET, *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, Springer Optimization and Its Applications, Springer New York, 2011, ch. Proximal Splitting Methods in Signal Processing, pp. 185–212.

[15] Q. DAI, C. FU, Y. JI, AND Y. ZHANG, *Robust joint reconstruction in compressed multi-view imaging*, in Proc. Picture Coding Symp., Poland, May 2012, pp. 13–16.

[16] M. F. DUARTE, S. SARVOTHAM, M. B. WAKIN, D. BARON, AND R. G. BARANIUK, *Joint sparsity models for distributed compressed sensing*, in Proc. Workshop on Signal Processing with Adaptive Sparse Structured Representations, 2005.

[17] M. J. FADILI AND J.-L. STARCK, *Monotone operator splitting for optimization problems in sparse recovery*, in IEEE Conf. Image Processing, Cairo, Egypt, 2009, pp. 1461 – 1464.

[18] S. FARSIU, D. ROBINSON, M. ELAD, AND P. MILANFAR, *Fast and robust multi-frame super-resolution*, IEEE Trans. Image Process., 12 (2004), pp. 1327–1344.

[19] C. FU, Y. JI, AND Q. DAI, *Compressed multi-view imaging with joint reconstruction*, in Proc. Data Compression Conf., 2011, p. 448.

[20] R. KEYS, *Cubic convolution interpolation for digital image processing*, IEEE Trans. Acoustics, Speech, Signal Process., 29 (1981).

[21] A. KUBOTA, A. SMOLIC, M. MAGNOR, M. TANIMOTO, T. CHEN, AND C. ZHANG, *Multiview imaging and 3DTV*, IEEE Signal Process. Mag., 24 (2007), pp. 10–21.

[22] X. LI, Z. WEI, AND L. XIAO, *Compressed sensing joint reconstruction for multi-view images*, Electronics Lett., 46 (2010), pp. 1548 – 1550.

[23] H. MAMAGHANIAN, N. KHALED, D. ATIENZA ALONSO, AND P. VANDERGHEYNST, *Design and exploration of low-power analog to information conversion based on compressed sensing*, IEEE J. Emerg. Sel. Topics in Circuits Syst., (2012).

[24] J. Y. PARK AND M. B. WAKIN, *A geometric approach to multi-view compressive imaging*, EURASIP J. Adv. Signal Process., 2012 (2012).

[25] Y. PENG, A. GANESH, J. WRIGHT, W. XU, AND Y. MA, *RASL: Robust Alignment by Sparse and Low-rank decomposition for linearly correlated images*, in IEEE Conf. Computer Vision and Pattern Recognition, 2010, pp. 763 – 770.

[26] ——, *RASL: Robust Alignment by Sparse and Low-rank decomposition for linearly correlated images*, IEEE Trans. Pattern Anal. Mach. Intell., 34 (2012), pp. 2233–2246.

[27] T. Q. PHAM, L. J. V. VLIET, AND K. SCHUTTE, *Robust fusion of irregularly sampled data using adaptive normalized convolution*, EURASIP J. Applied Signal Process., 2006 (2006).

[28] M. PROTTER, M. ELAD, H. TAKEDA, AND P. MILANFAR, *Generalizing the nonlocal-means to super-resolution reconstruction*, IEEE Trans. Image Process., 18 (2009), pp. 36–51.

[29] G. PUY AND P. VANDERGHEYNST, *Robust joint reconstruction of misaligned images using semi-parametric dictionaries*, in ICML Workshop on Sparsity, Dictionaries and Projections in Machine Learning and Signal Processing, Edinburgh, Scotland, 2012.

[30] ——, *Joint reconstruction of misaligned images from incomplete measurements for cardiac mri*, in International Conference on Sampling Theory and Applications (SAMPTA), Bremen, Germany, 2013.

[31] G. PUY, P. VANDERGHEYNST, R. GRIBONVAL, AND Y. WIAUX, *Universal and efficient compressed sensing by spread spectrum and application to realistic Fourier imaging techniques*, EURASIP J. Adv. Signal Process., 2012 (2012).

[32] H. RAGUET, J. FADILI, AND G. PEYRÉ, *A generalized forward-backward splitting*, SIAM J. Imaging Sci., to appear (2013).

[33] L. I. RUDIN, S. OSHER, AND E. FATEMI, *Nonlinear total variation based noise removal algorithms*, Physica

D: Nonlinear Phenomena, 60 (1992), pp. 259–268.

[34] A. C. SANKARANARAYANAN, P. K. TURAGA, R. G. BARANIUK, AND R. CHELLAPPA, *Compressive acquisition of dynamic scenes*, in Computer Vision - ECCV 2010, vol. 6311, 2010, pp. 129–142.

[35] M. SCHMIDT, D. KIM, AND S. SRA, *Optimization for Machine Learning*, MIT press, 2011, ch. Projected Newton-type methods in machine learning.

[36] H. SHEN, L. ZHANG, B. HUANG, AND P. LI, *A MAP approach for joint motion estimation, segmentation, and super resolution*, IEEE Trans. Image Process., 16 (2007), pp. 479–490.

[37] C. STEHNING, P. BÖRNERT, K. NEHRKE, H. EGGERS, AND M. STUBER, *Free-breathing whole-heart coronary MRA with 3D radial SSFP and self-navigated image reconstruction*, Magn. Reson. Med., 54 (2005), pp. 476–480.

[38] C. STRECHA, W. VON HANSEN, L. VAN GOOL, P. FUA, AND U. THOENNESSEN, *On benchmarking camera calibration and multi-view stereo for high resolution imagery*, in IEEE Conf. Computer Vision and Pattern Recognition, 2008, pp. 1 − 8.

[39] V. THIRUMALAI AND P. FROSSARD, *Joint reconstruction of multi-view compressed images*, IEEE Trans. Image Process., submitted, (2012).

[40] M. TROCAN, T. MAUGEY, J. E. FOWLER, AND B. PESQUET-POPESCU, *Disparity-compensated compressed-sensing reconstruction for multiview images*, in Proc. IEEE Conf. Multimedia and Expo, 2010, pp. 1225 − 1229.

[41] M. TROCAN, T. MAUGEY, E. W. TRAMEL, J. E. FOWLER, AND B. PESQUET-POPESCU, *Compressed sensing of multiview images using disparity compensation*, in Proc. IEEE Conf. Image Processing, 2010, pp. 3345 − 3348.

[42] P. TSENG, *Convergence of a block coordinate descent method for nondifferentiable minimization*, J. Optim. Theory Appl., 109 (2001), pp. 475–494.

[43] M. UNSER, *Sampling—50 years after Shannon*, Proc. IEEE, 88 (2000), pp. 569– 587.

[44] P. VANDEWALLE, S. SÜSSTRUNK, AND M. VERTELLI, *A frequency domain approach to registration of aliased images with application to super-resolution*, EURASIP J. Applied Signal Process., 2006 (2006), p. 71459.

[45] M. B. WAKIN, *A manifold lifting algorithm for multi-view compressive imaging*, in Proc. Picture Coding Symp., 2009, pp. 1 − 4.

[46] Z. ZHANG, A. GANESH, X. LIANG, AND Y. MA, *TILT: Transform Invariant Low-Rank Textures*, Int. J. Comput. Vis., 89 (2012), pp. 1–24.