RASL: Robust Alignment by Sparse and Low-rank Decomposition for Linearly Correlated Images

Yigang Peng, Arvind Ganesh, *Student Member, IEEE*,
John Wright, *Member, IEEE*, Wenli Xu, and Yi Ma, *Senior Member, IEEE*

Abstract

This paper studies the problem of simultaneously aligning a batch of linearly correlated images despite gross corruption (such as occlusion). Our method seeks an optimal set of image domain transformations such that the matrix of transformed images can be decomposed as the sum of a sparse matrix of errors and a low-rank matrix of recovered aligned images. We reduce this extremely challenging optimization problem to a sequence of convex programs that minimize the sum of ℓ^1-norm and nuclear norm of the two component matrices, which can be efficiently solved by scalable convex optimization techniques. We verify the efficacy of the proposed robust alignment algorithm with extensive experiments on both controlled and uncontrolled real data, demonstrating higher accuracy and efficiency than existing methods over a wide range of realistic misalignments and corruptions.

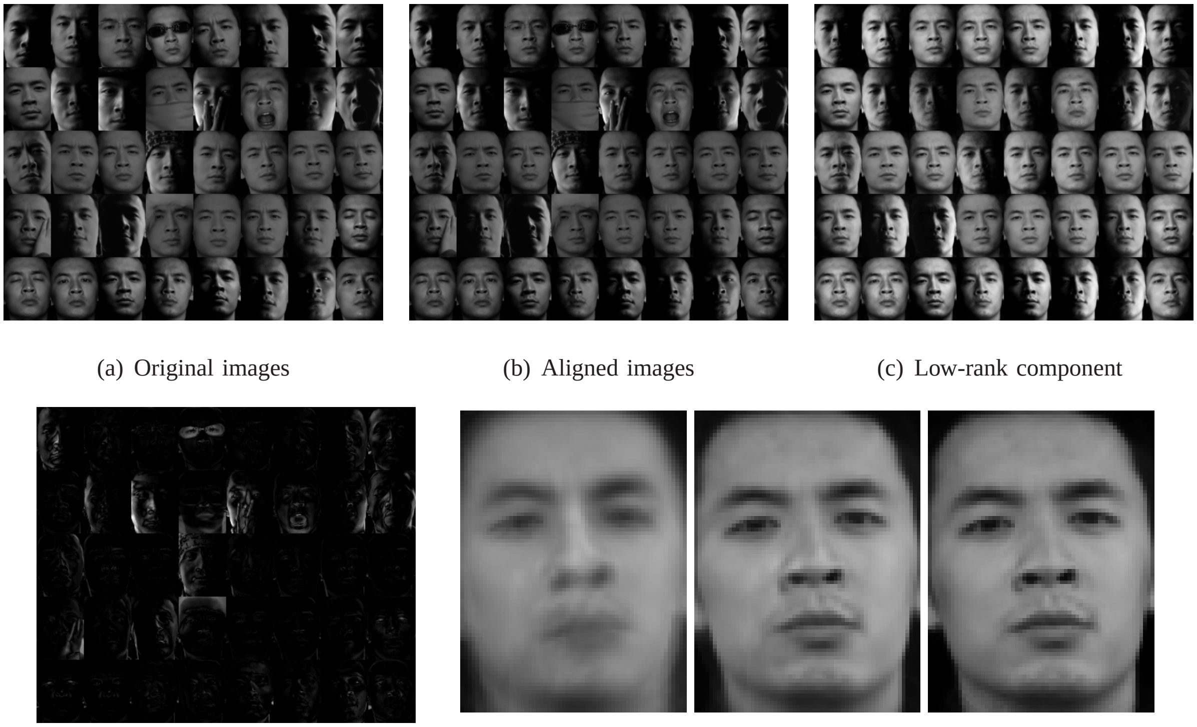
Index Terms

Batch Image Alignment, Rank Minimization, Principal Component Pursuit, Occlusion and Corruption.

A. Ganesh and Y. Ma are with the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign. J. Wright and Y. Ma are with Microsoft Research Asia. Y. Peng and W. Xu are with TNLIST and the Department of Automation, Tsinghua University. Corresponding author: Arvind Ganesh, 146 Coordinated Science Laboratory, 1308 W. Main St., Urbana, Illinois 61801. Email: abalasu2@illinois.edu. Tel: 1-217-244-9414.

I. Introduction

In recent years, the increasing popularity of image and video sharing sites such as Facebook, Flickr, and YouTube has led to a dramatic increase in the amount of visual data available online. Within the computer vision community, this has inspired a renewed interest in large, unconstrained datasets [1]. Such data pose steep challenges to existing vision algorithms: significant illumination variation, partial occlusion, as well as poor or even no alignment (see Figure 1(a) for example). This last difficulty is especially challenging, since domain transformations make it difficult to measure image similarity for recognition or classification. Intelligently harnessing the information encoded in these large sets of images seems to require more efficient and effective solutions to the long-standing batch image alignment problem [2], [3]: *Given many images of an object or objects of interest, align them to a fixed canonical template*.



(a) Original images

(b) Aligned images

(c) Low-rank component

(d) Recovered errors

(e) Average of (a), (b), (c), respectively

Fig. 1. **Batch Image Alignment.** (a) 40 face images of a person with different illumination, occlusions, poses, and expressions. Our algorithm automatically finds a set of transformations such that the transformed images in (b) can be decomposed as the sum of images from a low-rank approximation in (c) and sparse large errors in (d). The much sharpened average face images shown in (e) indicate the efficacy of our alignment algorithm.

To a large extent, progress in batch image alignment has been driven by the introduction of increasingly sophisticated measures of image similarity [4]. Learned-Miller's influential *congealing* algorithm seeks an alignment that minimizes the sum of entropies of pixel values at each pixel location in the batch of aligned images [5], [6]. If we stack the aligned images as the columns of a large matrix, this criterion demands that each row of this matrix be nearly constant. Conversely, the *least squares congealing* procedure of [7], [8] seeks an alignment that minimizes the sum of squared distances between pairs of images, and hence demands that the columns be nearly constant. In both cases, if the criterion is satisfied exactly, the matrix of aligned images will have *low rank*, ideally rank one. However, if there is large illumination variation in the images (such as those in Figure 1), the matrix of aligned images might have an *unknown* rank higher than one. In this case, it is more appropriate to search for an alignment that minimizes the rank of the aligned images. So in [9], Vedaldi et. al. choose to minimize a log-determinant measure that can be viewed as a smooth surrogate for the rank function [10]. The low-rank objective can also be directly enforced, as in *Transformed Component Analysis* (TCA) [11], [12], which uses an EM algorithm to fit a low-dimensional linear model, subject to domain transformations drawn from a known group.

A major drawback of the above approaches is that they do not simultaneously handle large illumination variations and gross pixel corruptions or partial occlusions that often occur in real images (e.g., shadows, hats, glasses in Figure 1). The *Robust Parameterized Component Analysis* (RPCA) algorithm of [13] also fits a low-rank model, and uses a robust fitting function to reduce the influence of corruption and occlusion. Unfortunately, this leads to a difficult, nonconvex optimization problem, with no theoretical guarantees of robustness or convergence rate. This somewhat unsatisfactory status quo is mainly due to the extremely difficult nature of the core problem of fitting a low-rank model to highly corrupted data [14], a problem that until recently lacked a polynomial-time algorithm with strong performance guarantees. Recent advances in rank minimization [15], [16] have shown that it is indeed possible to efficiently and exactly recover low-rank matrices despite significant corruption, using tools from convex programming. These developments prompt us to revisit the problem of robustly aligning batches of linearly correlated images.

Contributions. In this paper, we introduce a new algorithm, named RASL, for robustly aligning linearly correlated images (or signals), despite large occlusions and corruptions. Our solution builds on recent advances in rank minimization and formulates the batch alignment problem as the solution to a sequence of convex programs. We show how each of these convex programs can be solved efficiently using modern first-order optimization techniques, leading to a fast, scalable algorithm that succeeds under very broad conditions. Our algorithm can handle batches of over one hundred images in a few minutes on a standard PC. As we will verify with extensive experiments on real image data, the algorithm achieves pixelwise accuracy over a wide range of misalignments. A MATLAB implementation of our algorithm and the data used in this paper is publicly available on our website:

`http://perception.csl.uiuc.edu/matrix-rank/rasl.html`.

Organization. The remainder of this paper is organized as follows: In Section II, we introduce matrix rank as a measure of image similarity and recast the image alignment problem as one of matrix rank minimization. In Section III, we propose an efficient algorithm to solve the rank minimization problem by iterative convex optimization. We provide experimental results in Section IV to showcase the efficacy of our method on real images. Section V provides concluding remarks and proposes potential extensions to our algorithm.

II. Image Alignment by Matrix Rank Minimization

In this section, we formulate batch image alignment as the search for a set of transformations that minimizes the rank of the transformed images, viewed as the columns of a matrix. We discuss why rank is a natural measure of image similarity, and how this conceptual framework can be made robust to gross errors due to corruption or occlusion.

A. Matrix Rank as a Measure of Image Similarity

Measuring the amount of similarity within a set of images is a fundamental problem in computer vision and image processing. Suppose we are given n well-aligned grayscale images $I_1^0, \ldots, I_n^0 \in \mathbb{R}^{w \times h}$ of some object or scene. In many situations of interest, these well-aligned images are *linearly correlated*. More precisely, if we let $\mathrm{vec} : \mathbb{R}^{w \times h} \to \mathbb{R}^m$ denote the operator that selects an m-pixel region of interest (typically $m \gg n$) from an image and stacks it as a vector, then as a matrix

$$A \doteq \left[\mathrm{vec}(I_1^0) \mid \cdots \mid \mathrm{vec}(I_n^0)\right] \in \mathbb{R}^{m \times n} \tag{1}$$

should be approximately *low-rank*. This assumption holds quite generally. For example, if the $I_i^0, i = 1, \ldots, n$ are images of some convex Lambertian object under varying illumination, then a rank-9 approximation suffices [17]. Being able to correctly identify this low-dimensional structure is crucial for many vision tasks such as face recognition.

B. Modeling Misalignments as Domain Deformations

Misalignment poses a serious problem to many different computer vision applications. It is an inherent problem in most image acquisition processes since the relative position of the camera with respect to the object is seldom fixed across multiple images. Images of the same object or scene can appear drastically different even under moderate change in the object's position or pose with respect to the camera. The above model (low-rank matrix of correlated images) breaks down if the images are even slightly misaligned with respect to each other.

In this work, since the 3-D structure of the object of interest is unknown, we assume that the misalignment is restricted to the image plane.[1] Then, we can model misalignments as domain deformations. More precisely, if I_1 and I_2 represent two misaligned images, then there exists an invertible transformation $\tau : \mathbb{R}^2 \to \mathbb{R}^2$ such that

$$I_2(x, y) = (I_1 \circ \tau)(x, y) \doteq I_1(\tau(x, y)). \tag{2}$$

In most practical scenarios, we can model misalignments as transformations from a finite-dimensional group $\mathbb{G}$ that has a parametric representation, such as the similarity group $SE(2) \times \mathbb{R}_+$, the 2-D affine group Aff(2), and the planar homography group $GL(3)$ (see [18] for more details on transformation groups).

Consolidating the above two models, we formulate the image alignment problem as follows. Suppose that $I_1, I_2, \ldots, I_n$ represent n input images of the same object but misaligned with respect to each other. Then, there exist domain transformations $\tau_1, \tau_2, \ldots, \tau_n$ such that the transformed images $I_1 \circ \tau_1, I_2 \circ \tau_2, \ldots, I_n \circ \tau_n$ are well-aligned at the pixel level, or equivalently the matrix

$$D \circ \tau \doteq \left[\mathrm{vec}(I_1^0) \mid \cdots \mid \mathrm{vec}(I_n^0)\right] \in \mathbb{R}^{m \times n}$$

[1] We will see in Section IV that the proposed algorithm is robust to small changes in 3-D pose as well.

has low rank, where $I_j^0 = I_j \circ \tau_j$ for $j = 1, 2, \ldots, n$, $D = [\mathrm{vec}(I_1) \mid \cdots \mid \mathrm{vec}(I_n)]$, and τ represents the set of n transformations $\tau_1, \tau_2, \ldots, \tau_n$. Therefore, the batch image alignment problem can be reduced to the following optimization problem:

$$\min_{A,\tau} \ \mathrm{rank}(A) \quad \mathrm{s.t.} \quad D \circ \tau = A. \tag{3}$$

C. Modeling Corruption and Occlusion as Large, Sparse Errors

In practice, the low-rank structure of the aligned images can be easily violated, due to the presence of partial occlusions or corruptions in the images. Since these errors typically affect only a small fraction of all pixels in an image, we can model them as sparse errors whose nonzero entries can have arbitrarily large magnitude. This model has been successfully employed in face recognition [19].

In addition to occlusions, real images typically contain some noise of small magnitude in each pixel. To keep our discussion simple, we assume here that such noise is negligible in magnitude as compared to the error due to occlusions. We will see in Section III-B that it is straightforward to incorporate this small-magnitude noise into our algorithm.

Let e_j represent the error corresponding to image I_j such that the images $\{I_j \circ \tau_j - e_j\}_{j=1}^n$ are well-aligned to each other, and free of any corruptions or occlusions. Therefore, the formulation (3) can be modified as follows:

$$\min_{A,E,\tau} \ \mathrm{rank}(A) \quad \mathrm{s.t.} \quad D \circ \tau = A + E, \ \ \|E\|_0 \leq k, \tag{4}$$

where $E = [\mathrm{vec}(e_1) \mid \cdots \mid \mathrm{vec}(e_n)]$. Here, the ℓ^0-“norm” $\|\cdot\|_0$ counts the number of nonzero entries in the error matrix E, and k is a constant that represents the maximum number of corrupted pixels expected across all images. As we will see in Section III-A, it is more convenient to consider the Lagrangian form of this problem:

$$\min_{A,E,\tau} \ \mathrm{rank}(A) + \gamma\|E\|_0 \quad \mathrm{s.t.} \quad D \circ \tau = A + E, \tag{5}$$

where $\gamma > 0$ is a parameter that trades off the rank of the solution versus the sparsity of the error. We refer to this problem as *Robust Alignment by Sparse and Low-rank decomposition* (RASL).

To summarize our approach (5) to solving the image alignment problem, we know that if the images are well-aligned, they should exhibit good low-rank structure, up to some sparse errors (say due to occlusions). We therefore search for a set of transformations $\tau = \{\tau_1, \ldots, \tau_n\}$ such that the rank of the transformed images becomes as small as possible, when the sparse errors are subtracted.

III. Practical Solution via Iterative Convex Programming

In this section, we present a practical solution to the RASL problem (5), that works quite effectively as long as the misalignments are not too large. We first relax the highly nonconvex objective function in (5) to its convex surrogate (Section III-A). We then linearize the nonlinear equality constraint in (5) (Section III-B), yielding a sequence of convex programs whose solutions converge quadratically to the correct alignment (Section III-C). These convex programs can be solved efficiently via modern first-order optimization techniques (Section III-D). In Section IV we will verify the practical convergence behavior of this scheme with numerous real-data examples.

A. Convex Relaxation

The optimization problem (5), although intuitive, is not directly tractable. A major difficulty is the nonconvexity of the matrix rank and ℓ^0-norm: minimization of these functions is extremely difficult (NP-hard and hard to approximate) in the worst case. Moreover, since matrix rank and the ℓ^0-norm are discrete-valued functions, the solution given by (5) is likely to be unstable if the errors in the images are not *exactly sparse*. Recently, however, it was shown that for the problem of recovering low-rank matrices from sparse errors, as long as the rank of the matrix A to be recovered is not too high and the number of non-zero entries in E is not too large, minimizing the natural convex surrogate for $\mathrm{rank}(A) + \lambda\|E\|_0$ can *exactly* recover A [16].[2] This convex relaxation replaces $\mathrm{rank}(\cdot)$ with the *nuclear norm* or sum of the singular values: $\|A\|_* \doteq \sum_{i=1}^{\min\{m,n\}} \sigma_i(A)$, and replaces the ℓ^0-norm $\|E\|_0$ with the ℓ^1-norm: $\sum_{ij} |E_{ij}|$. Applying the same relaxation to the RASL problem (5) yields a new optimization problem:

$$\min_{A,E,\tau} \|A\|_* + \lambda\|E\|_1 \quad \text{s.t.} \quad D \circ \tau = A + E. \tag{6}$$

[2] Convex programming exactly recovers low-rank matrices A whose singular vectors are not themselves sparse or spiky. More precisely, it succeeds with high probability (assuming that the support of E is random) provided $\mathrm{rank}(A) < C_1\mu^{-1}n/\log^2(m)$ and $\|E\|_0 < C_2 mn$, where C_1, C_2 are numerical constants and μ is an *incoherence* parameter that is small if the singular spaces of A are not aligned with the standard basis [16]. Similar guarantees can be proved for the linearized convex optimization to be introduced in Section III-D, but are not the main focus of this paper.

Theoretical considerations in [16] suggest that the weighting parameter λ should be of the form $C/\sqrt{m}$ where C is a constant, typically set to unity. The new objective function is non-smooth, but now continuous and convex.

B. Iterative Linearization

The main remaining difficulty in solving (6) is the nonlinearity of the constraint $D \circ \tau = A + E$, which arises due to the complicated dependence of $D \circ \tau$ on the transformations $\tau \in \mathbb{G}^n$. When the change in τ is small, we can approximate this constraint by linearizing about the current estimate of τ. Here, and below, we assume that $\mathbb{G}$ is some p-parameter group and identify $\tau = [\tau_1 \mid \cdots \mid \tau_n] \in \mathbb{R}^{p \times n}$ with the parameterizations of all of the transformations. For $\Delta\tau = [\Delta\tau_1 \mid \cdots \mid \Delta\tau_n] \in \mathbb{R}^{p \times n}$, write $D \circ (\tau + \Delta\tau) \approx D \circ \tau + \sum_{i=1}^{n} J_i \Delta\tau_i \epsilon_i^T$, where $J_i \doteq \frac{\partial}{\partial \zeta} \mathrm{vec}(I_i \circ \zeta)|_{\zeta = \tau_i} \in \mathbb{R}^{m \times p}$ is the Jacobian of the i-th image with respect to the transformation parameters τ_i and $\{\epsilon_i\}$ denotes the standard basis for $\mathbb{R}^n$. This leads to a convex optimization problem in unknowns $A, E, \Delta\tau$:

$$\min_{A,E,\Delta\tau} \|A\|_* + \lambda \|E\|_1 \ \text{ s.t. } \ D \circ \tau + \sum_{i=1}^{n} J_i \Delta\tau \epsilon_i \epsilon_i^T = A + E. \tag{7}$$

Because the linearization only holds locally, we should not expect the solution $\tau + \Delta\tau$ from (7) to exactly solve (6). To find the (local) minimum of (6), we repeatedly linearize about our current estimate of τ and solve a sequence of convex programs of the form (7).[3] As we will see in Section IV, as long as the initial misalignment is not too large, this iteration effectively recovers the correct transformations τ and separates the low-rank structure of the batch of images from any sparse errors or occlusions. This complete optimization procedure is summarized as Algorithm 1. The iterative procedure in Algorithm 1 is stopped when the relative change in the value of the cost function between two consecutive iterations is smaller than a pre-determined threshold. Notice that Algorithm 1 operates on the normalized images $\mathrm{vec}(I_i \circ \tau_i)/\|\mathrm{vec}(I_i \circ \tau_i)\|_2$, in order to rule out trivial solutions such as zooming in on a single dark pixel or a dark region in the images.

[3] This kind of iterative linearization has a long history in gradient algorithms for batch image alignment (see, e.g., [9], [20] and references therein). More recently a similar iterative convex programming approach was proposed for single-to-batch image alignment in face recognition [21].

Algorithm 1 (Outer loop of RASL)

```
  INPUT: Images $I_1, \ldots, I_n \in \mathbb{R}^{w \times h}$, initial transformations $\tau_1, \ldots, \tau_n$ in a certain parametric group $\mathbb{G}$, weight $\lambda > 0$.
  WHILE not converged DO
      Step 1: compute Jacobian matrices w.r.t. transformation:
          $J_i \leftarrow \frac{\partial}{\partial \zeta} \left( \frac{\mathrm{vec}(I_i \circ \zeta)}{\|\mathrm{vec}(I_i \circ \zeta)\|_2} \right) \Big|_{\zeta = \tau_i}, \quad i = 1, \ldots, n;$
      Step 2: warp and normalize the images:
          $D \circ \tau \leftarrow \left[ \frac{\mathrm{vec}(I_1 \circ \tau_1)}{\|\mathrm{vec}(I_1 \circ \tau_1)\|_2} \Big| \ldots \Big| \frac{\mathrm{vec}(I_n \circ \tau_n)}{\|\mathrm{vec}(I_n \circ \tau_n)\|_2} \right];$
      Step 3 (inner loop): solve the linearized convex optimization:
          $(A^*, E^*, \Delta\tau^*) \leftarrow \arg\min_{A, E, \Delta\tau} \quad \|A\|_* + \lambda \|E\|_1$
          $\text{s.t.} \ \ D \circ \tau + \sum_{i=1}^{n} J_i \Delta\tau \epsilon_i \epsilon_i^T = A + E;$
      Step 4: update transformations: $\tau \leftarrow \tau + \Delta\tau^*$;
  END WHILE
  OUTPUT: solution $A^*$, $E^*$, $\tau^*$ to problem (6).
```

In this work, we have considered only sparse, large-magnitude errors in images arising from occlusions or other forms of corruption. However, in practice, images also contain some noise of small magnitude in each pixel. This can be easily augmented into our model by adding a *noise matrix* Z of bounded magnitude to the equality constraint in (7). It has been shown in [22] that sparse and low-rank matrix decomposition (without transformations) by convex optimization is stable to additive Gaussian noise of small magnitude, in addition to sparse errors. It may be possible to establish similar stability guarantees for the linearized convex program in (7). We defer this to future work since it is beyond the scope of this paper.

C. Convergence and Optimality

Replacing a difficult optimization problem with a sequence of more tractable, linearized problems is a standard technique in optimization, and has been the subject of intensive study in the optimization literature. As we will see, the RASL algorithm can be viewed as a Gauss-

Newton method for minimizing the composition of a nonsmooth convex function with a smooth, nonlinear mapping. The convergence behavior of such algorithms was extensively studied in the late 1970's and early 1980's, and they continue to draw attention today [23]. We draw upon this body of work, in particular results of Jittorntrum and Osborne [24] (building on work of Cromme [25]) to understand the local convergence of RASL.

The result of [24] concerns the problem of minimizing the composition of a norm $\|\cdot\|_\diamond : \mathbb{R}^n \to \mathbb{R}$ with a C^2 mapping $f : \mathbb{R}^p \to \mathbb{R}^n$:

$$\min_{x \in \mathbb{R}^p} \; \|f(x)\|_\diamond, \tag{8}$$

The authors of [25], [24] have studied the iterative algorithm

$$\begin{aligned} \delta_k &= \arg\min_{\delta \in \mathbb{R}^p} \left\| f(x_k) + \frac{\partial f}{\partial x}(x_k)\delta \right\|_\diamond, & (9) \\ x_{k+1} &= x_k + \delta_k, & (10) \end{aligned}$$

and have shown that if $x^\star \in \mathbb{R}^p$ is a *strictly unique* optimum to (8), in the sense that $\exists\, \alpha > 0$ such that

$$\forall\, \delta \in \mathbb{R}^p, \qquad \left\| f(x^\star) + \frac{\partial f}{\partial x}(x^\star)\delta \right\|_\diamond \geq \|f(x^\star)\|_\diamond + \alpha\|\delta\|, \tag{11}$$

then within some neighborhood of $x^\star$, the sequence of iterates (9)-(10) converges quadratically to $x^\star$.

To clarify the connection to RASL, we define a function $\|\cdot\|_\diamond : \mathbb{R}^{m\times n} \to \mathbb{R}$ via

$$\|M\|_\diamond \doteq \min_{A+E=M} \|A\|_* + \lambda\|E\|_1. \tag{12}$$

It is easy to verify that $\|\cdot\|_\diamond$ is indeed a norm[4] – it is a quotient norm on translates of $\{(-X, X) \mid X \in \mathbb{R}^{m\times n}\} \subset \mathbb{R}^{m\times n} \times \mathbb{R}^{m\times n}$. Let the transformations $\tau = \{\tau_1 \ldots \tau_n\}$ be parameterized by parameters $x = \{\xi_1 \ldots \xi_n\} \in (\mathbb{R}^p)^n$. Then, we can write $D \circ \tau = f(x)$, and view the RASL optimization as a local procedure for solving the problem

$$\min_{x \in (\mathbb{R}^p)^n} \|f(x)\|_\diamond, \tag{13}$$

via the iteration (9)-(10). Hence, provided the map $x \mapsto f(x)$ is C^2, the result of [24] implies that RASL converges quadratically in the neighborhood of any strongly unique local minimum.

[4] It is easy to check that $\|M\|_\diamond \geq 0$ with equality iff $M = 0$, and that $\|tM\|_\diamond = |t|\|M\|_\diamond$. The triangle inequality follows from the convexity of the function $\|A\|_* + \lambda\|E\|_1$.

This quadratic convergence is observed in our experiments (see Section IV), in which only a few iterations, typically less than 20, are required for the algorithm to converge.

It is important to realize that, in general, manifolds formed by transformed images may not be C^2 (or not even C^1), due to the presence of sharp edges [26]. However, in our case, we can view the digital images $I_i \circ \tau_i$ as resampling transformations of an ideal bandlimited reconstruction $\hat{I}_i$ obtained from the digital image I_i, in which case the mapping $x \mapsto f(x)$ is indeed smooth.

A complete convergence theory would then verify, based on the properties of the desired solution $x^\star$, that the strong uniqueness property (11) holds. It is not difficult to give quantitative bounds on the region of convergence and convergence rate of the algorithm based on the coefficient α in (11) and the curvature of the set $\{f(x) \mid x \in (\mathbb{R}^p)^n\} \subset \mathbb{R}^{m\times n}$. However, estimating α or characterizing the curvature are themselves nontrivial mathematical problems, which we delay to future work. The interested reader may consult [22], where a form of strong uniqueness is (implicitly) used to show the stability of sparse and low-rank decomposition, albeit without transformations.

D. Efficient Solution by Augmented Lagrange Multiplier Methods

The main computational cost in Algorithm 1 at each iteration is step 3, which solves the linearized convex optimization problem (7). This is a semidefinite program in thousands or millions of variables, so scalable solutions are essential for its practical use. Fortunately, a recent flurry of work on high-dimensional nuclear norm minimization has shown that such problems are well within the capabilities of a standard PC [27], [28], [29]. In this section, we show how one such fast first-order method, the Augmented Lagrange Multiplier (ALM) algorithm [29], [30], [16], can be adapted to efficiently solve (7).

The basic idea of the ALM method is to search for a saddle point of the augmented Lagrangian function instead of directly solving the original constrained optimization problem. Let us define $h(A, E, \Delta\tau) = D \circ \tau + \sum_{i=1}^n J_i \Delta\tau \epsilon_i \epsilon_i^T - A - E$. For our problem (7), the augmented Lagrangian function is given by

$$\mathcal{L}_\mu(A, E, \Delta\tau, Y) = \|A\|_* + \lambda\|E\|_1 + \langle Y, h(A, E, \Delta\tau)\rangle + \frac{\mu}{2}\|h(A, E, \Delta\tau)\|_F^2, \qquad (14)$$

where $Y \in \mathbb{R}^{m\times n}$ is a Lagrange multiplier matrix, μ is a positive scalar, $\langle \cdot, \cdot \rangle$ denotes the matrix inner product,[5] and $\|\cdot\|_F$ denotes the Frobenius norm. For appropriate choice of the Lagrange multiplier matrix Y and sufficiently large constant μ, it can be shown that the augmented Lagrangian function has the same minimizer as the original constrained optimization problem [30]. The ALM algorithm iteratively estimates both the Lagrange multiplier and the optimal solution by iteratively minimizing the augmented Lagrangian function

$$\begin{aligned}(A_{k+1}, E_{k+1}, \Delta\tau_{k+1}) &= \arg\min_{A,E,\Delta\tau} \mathcal{L}_{\mu_k}(A, E, \Delta\tau, Y_k),\\ Y_{k+1} &= Y_k + \mu_k\, h(A_{k+1}, E_{k+1}, \Delta\tau_{k+1}).\end{aligned} \tag{15}$$

It has been shown that when $\{\mu_k\}$ is a monotonically increasing positive sequence, the iteration indeed converges to the optimal solution of the problem (7) [30].

However, the first step in the above iteration (15) is difficult to solve directly. So typically, people choose to to minimize the Lagrangian function *approximately* by adopting an alternating strategy: minimize the function against the three unknowns $A, E, \Delta\tau$ one at a time:

$$\begin{aligned}A_{k+1} &= \arg\min_{A} \mathcal{L}_{\mu_k}(A, E_k, \Delta\tau_k, Y_k),\\ E_{k+1} &= \arg\min_{E} \mathcal{L}_{\mu_k}(A_{k+1}, E, \Delta\tau_k, Y_k),\\ \Delta\tau_{k+1} &= \arg\min_{\Delta\tau} \mathcal{L}_{\mu_k}(A_{k+1}, E_{k+1}, \Delta\tau, Y_k).\end{aligned} \tag{16}$$

Although each step of the above iteration involves solving a convex program, each has a simple closed-form solution, and hence, can be solved efficiently by a single step. To spell out the solutions, let us define the *soft-thresholding* or *shrinkage* operator for scalars as follows:

$$\mathcal{S}_\alpha[x] = \mathrm{sign}(x)\cdot\max\{|x| - \alpha, 0\}, \tag{17}$$

where $\alpha \geq 0$. When applied to vectors and matrices, the shrinkage operator acts elementwise. Using the shrinkage operator, we can write the solution to each step of (16) as

$$\begin{aligned}(U, \Sigma, V) &= \mathrm{svd}\left(D\circ\tau + \textstyle\sum_{i=1}^n J_i\Delta\tau_k\epsilon_i\epsilon_i^T + \frac{1}{\mu_k}Y_k - E_k\right),\\ A_{k+1} &= U\mathcal{S}_{\frac{1}{\mu_k}}[\Sigma]V^T,\\ E_{k+1} &= \mathcal{S}_{\frac{\lambda}{\mu_k}}\left[D\circ\tau + \textstyle\sum_{i=1}^n J_i\Delta\tau_k\epsilon_i\epsilon_i^T + \frac{1}{\mu_k}Y_k - A_{k+1}\right],\\ \Delta\tau_{k+1} &= \textstyle\sum_{i=1}^n J_i^\dagger\left(A_{k+1} + E_{k+1} - D\circ\tau - \frac{1}{\mu_k}Y_k\right)\epsilon_i\epsilon_i^T,\end{aligned} \tag{18}$$

where $\mathrm{svd}(\cdot)$ denotes the Singular Value Decomposition operator, and $J_i^\dagger$ denotes the Moore-Penrose pseudoinverse of J_i. For completeness, the entire algorithm to solve the linearized inner loop (7) has been summarized as Algorithm 2.

[5] $\langle X, Y\rangle \doteq \mathrm{trace}(X^TY)$.

Algorithm 2 (Inner Loop of RASL)

```
INPUT: $(A^0, E^0, \Delta\tau^0) \in \mathbb{R}^{m\times n} \times \mathbb{R}^{m\times n} \times \mathbb{R}^{p\times n}$, $\lambda > 0$.
WHILE not converged DO
    $(U, \Sigma, V) = \operatorname{svd}\big(D \circ \tau + \sum_{i=1}^{n} J_i \Delta\tau_k \epsilon_i \epsilon_i^T + \frac{1}{\mu_k} Y_k - E_k\big);$
    $A_{k+1} = U \mathcal{S}_{\frac{1}{\mu_k}}[\Sigma] V^T;$
    $E_{k+1} = \mathcal{S}_{\frac{\lambda}{\mu_k}}\big[D \circ \tau + \sum_{i=1}^{n} J_i \Delta\tau_k \epsilon_i \epsilon_i^T + \frac{1}{\mu_k} Y_k - A_{k+1}\big];$
    $\Delta\tau_{k+1} = \sum_{i=1}^{n} J_i^{\dagger}\big(A_{k+1} + E_{k+1} - D \circ \tau - \frac{1}{\mu_k} Y_k\big) \epsilon_i \epsilon_i^T;$
    $Y_{k+1} = Y_k + \mu_k \, h(A_{k+1}, E_{k+1}, \Delta\tau_{k+1}).$
END WHILE
OUTPUT: solution $(A^*, E^*, \Delta\tau^*)$ to problem (7).
```

In our experience, the algorithm always converges to the optimal solution to (7), and does so significantly faster than other alternative convex optimization methods. In particular, it is about 5-10 times faster than the accelerated proximal gradient (APG) method originally proposed in the conference version of this work [31]. Although the convergence of the ALM method (15) has been well established in the optimization literature, we currently know of no proof that its approximation (16) converges too. The main difficulty comes from the fact that there are three terms in the alternating minimization. The case with alternating between two terms has been studied extensively as the *alternating direction method of multipliers* in the optimization literature and its convergence has been well established for various cases [32], [33], [34]. In particular, the convergence for the Principal Component Pursuit problem – essentially problem (7) without the term associated with $\Delta\tau$ – has been established in [29]. Recently, [35] obtained a convergence result for certain three-term alternation applied to the noisy principal component pursuit problem (see also [36]). However, [35] reflects a very similar theory-practice gap – the three-term alternation for which convergence has been established is slower in practice than an alternation in the form of algorithm (18), for which a rigorous proof of convergence remains elusive.

E. Implementation details

In this section, we provide some details of our implementation of Algorithm 2 that could of be of help to practitioners in similar fields. For our experiments, we choose $\mu_k = \rho^k \mu_0$, where ρ and μ_0 are set to 1.25 and $1.25/\|D\|$, respectively.[6] The algorithm is terminated when the difference in the value of the cost function is sufficiently small between two consecutive iterations.

A minor practical issue with our algorithm is that poorly conditioned Jacobian matrices J_i's could lead to problems with numerical precision. Hence, we do not use them directly in Algorithm 2. Instead, we compute the QR factorization of the J_i's as $J_i = Q_i R_i^T$, and use the orthogonal Q_i's in Algorithm 2 in the place of the corresponding J_i's. This, in turn, implies that the output of the algorithm would be $\Delta\tau_i' = R_i \Delta\tau_i$. Since the R_i's are invertible, the change in the original deformation parameters $\Delta\tau_i$'s can be easily computed. Although this does not affect the theoretical convergence of the algorithm, we observe that it leads to a more stable implementation in practice.

IV. Experimental Verification

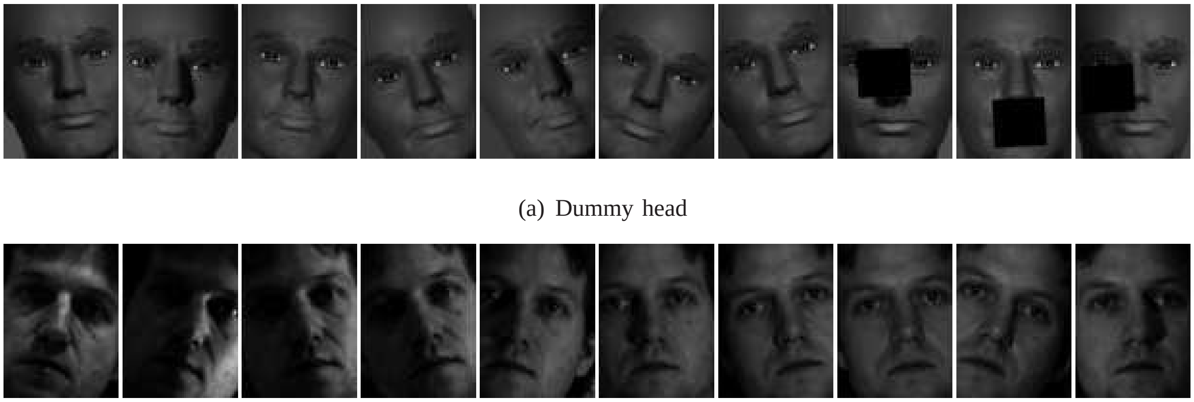
In this section, we demonstrate the efficacy of RASL on a variety of image alignment tasks. We always set $\lambda = 1/\sqrt{m}$ in the RASL algorithm, where m is the number of pixels in the region of interest in each image.[7] We first quantitatively verify the correctness of our algorithm on controlled data sets, and show that it outperforms state-of-the-art methods in aligning batches of images despite lighting variation and occlusion. We then test our algorithm on more realistic and challenging face images taken from the Labeled Faces in the Wild (LFW) database [1]. Experiments on video data, microscopic iris images, and handwritten digits further demonstrate the generality and broad applicability of our method. Finally, we provide an example of aligning perspective images of a planar surface that demonstrates its ability to cope with more complicated deformations such as planar homographies.

A. Quantitative Validation with Controlled Images

We verify the correctness of the algorithm using 100 images of a dummy head taken under varying illumination. Because the relative position between the camera and the dummy is fixed,

[6] $\|\cdot\|$ denotes the matrix spectral norm.

[7] The only exception in this paper is Figure 1, where we set $\lambda = 1.1/\sqrt{m}$.



(a) Dummy head

(b) Multi-PIE database

Fig. 2. **Sample input images.** Representative input images taken under controlled conditions artificially perturbed and occluded.

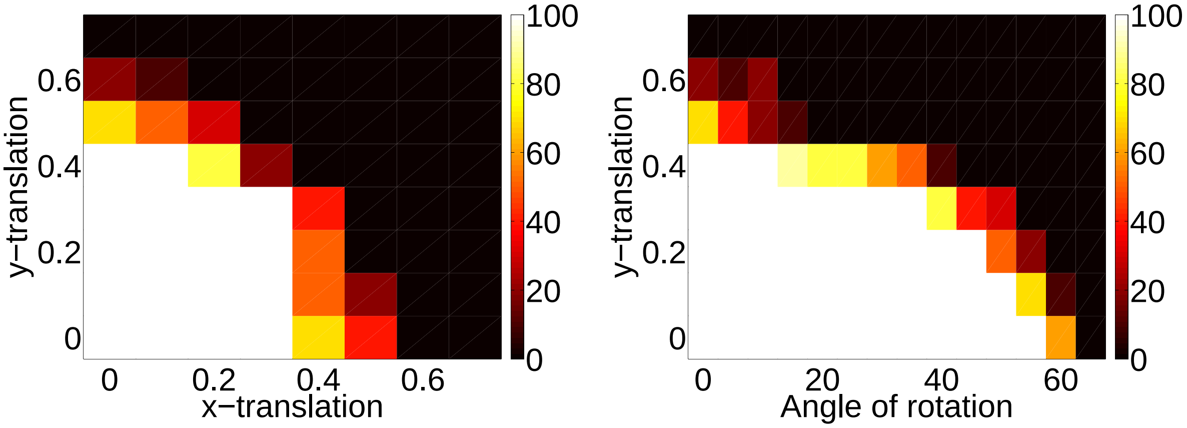
the ground truth alignment is known. We also test our algorithm on the CMU Multi-PIE face database [37] to illustrate its performance on more natural face images taken under controlled conditions. Figure 2 shows some representative sample input images used for our experiments.

1) Large region of attraction for RASL: We examine RASL's ability to cope with varying levels of misalignment. The task is to align the images to an 80×60 pixel canonical frame, in which the distance between the outer eye corners is normalized to 50 pixels.[8] We synthetically perturb each of the input images by Euclidean transformations ($\mathbb{G} = SE(2)$) whose angles of rotation are uniformly distributed in the range $[-\theta_0/2, \theta_0/2]$, and whose x- and y-translations are uniformly distributed in the range $[-x_0/2, x_0/2]$ pixels and $[-y_0/2, y_0/2]$ pixels, respectively.

We consider an alignment successful if the *maximum* difference in each individual coordinate of the eye corners across all pairs of images is less than one pixel in the canonical frame. Figure 3(a) shows the fraction of successes over 10 independent trials, with $\theta_0 = 0$ fixed and varying levels of translation x_0, y_0. Our algorithm always correctly aligns the images as long as x_0 and y_0 are each smaller than 15 pixels, i.e. 30% of the distance between the eyes. In Figure 3(b), we fix $x_0 = 0$ and plot the fraction of successful trials while varying both y_0 and θ_0. Here, RASL successfully aligns the given images despite translations of up to 15 pixels and simultaneous in-plane rotation of up to $40°$!

We repeat the above experiment with images of 100 subjects (users 001-100) chosen from

[8] The outer eye corners were manually chosen for one image, and the same set of coordinates were used for all images.



(a) Translation in x and y directions

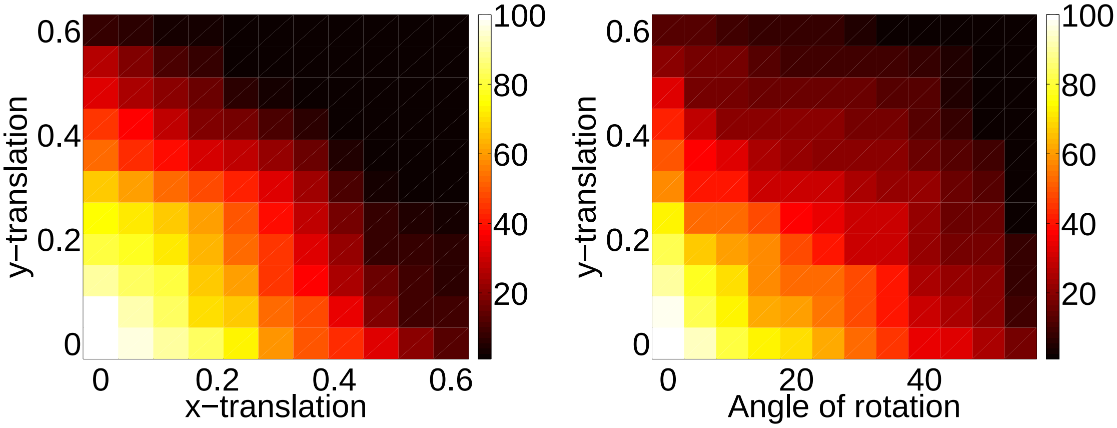
(b) Translation in y direction and in-plane rotation θ

Fig. 3. **Large region of attraction for RASL.** Percentage of successful alignments for varying levels of misalignment. Translations are given as a fraction of the distance between the eyes (here, 50 pixels), while rotations are in degrees. (a) Translation in x and y directions. All images are correctly aligned despite simultaneous x and y translations up to 30% of the eye distance. (b) Translation in y direction and in-plane rotation θ (degrees). All images are correctly aligned for despite simultaneous y translation of 30% of the eye distance and rotation up to $40°$.

Session 1 of the Multi-PIE database. The database contains 20 images of each subject taken under different illumination conditions. We once again use manually clicked outer eye corners to crop the images. This set of images is much more challenging than in the previous experiment since we have only 20 images per person. For each subject, we consider one instance of a randomly chosen misalignment as described above, and record the percentage of successful alignments across all subjects. The experimental results are shown in Figure 4. We notice that RASL achieves a success rate of over 90% even when there's simultaneous misalignment in both x and y directions of about 7 pixels.

2) *Effect of number of images:* It is clear that the region of attraction for the Multi-PIE images (Figure 4) is smaller than that for the dummy head images (Figure 3). A primary reason for this difference is the fact that the Multi-PIE database contains only 20 images per person, as against 100 images of the dummy head. In this experiment, we study the effect of the number of images on the region of attraction.

We use the 100 images of a dummy head described earlier. We choose subsets of images from this dataset, and artificially perturb them in the same manner as was done for the region of attraction experiment (see Figure 3). In this experiment, we perturb the images only along the



(a) Translation in x and y directions

(b) Translation in y direction and in-plane rotation θ

Fig. 4. **Region of attraction for RASL with Multi-PIE images.** Percentage of successful alignments for varying levels of misalignment. Translations are given as a fraction of the distance between the eyes (here, 50 pixels), while rotations are in degrees. (a) Translation in x and y directions. (b) Translation in y direction and in-plane rotation θ (degrees).

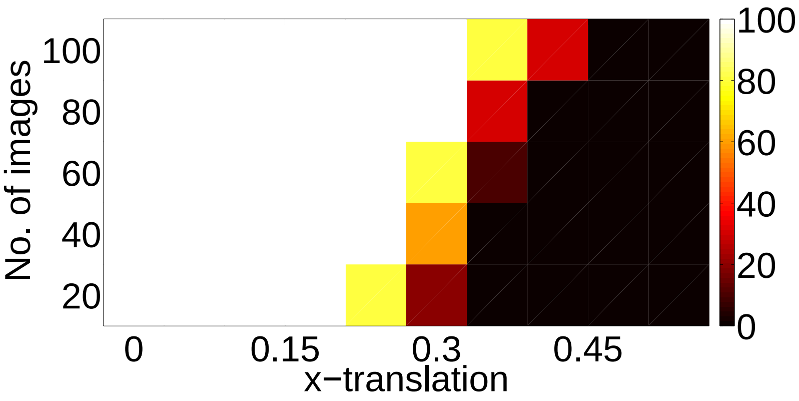


Fig. 5. **Effect of number of images on region of attraction.** Percentage of successful alignments for varying levels of misalignment and number of images. The misalignment is restricted to translation along the x-direction. The region of attraction steadily increases as the number of images is increased.

x-direction, where each image is translated by an amount uniformly distributed in the interval $[-x_0/2, x_0/2]$ pixels. Figure 5 summarizes the results of this experiment where the success rate has been measured over 10 independent trials. We observe that the region of attraction increases as the number of images increases. This is because with more images, the redundancy in the data is higher and hence, the low-rank model fits better.

3) Handling occlusion: A major advantage of the formulation of RASL is that it can handle large magnitude corruption, like occlusions, in the input images. For practical applications, it is interesting to know beforehand the amount of occlusion that RASL can handle for a given set of images. Unfortunately, this is very hard to characterize analytically since it depends on many factors, including the number of images, the amount of misalignment, the extent of linear correlation between the images, etc. In this experiment, we provide an empirical characterization of the amount of occlusion that RASL can handle for different levels of misalignments.

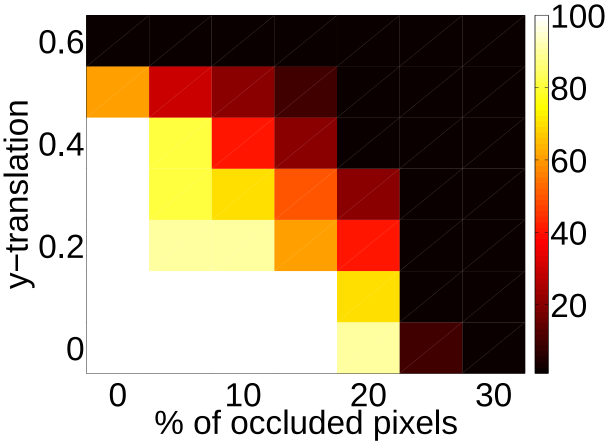
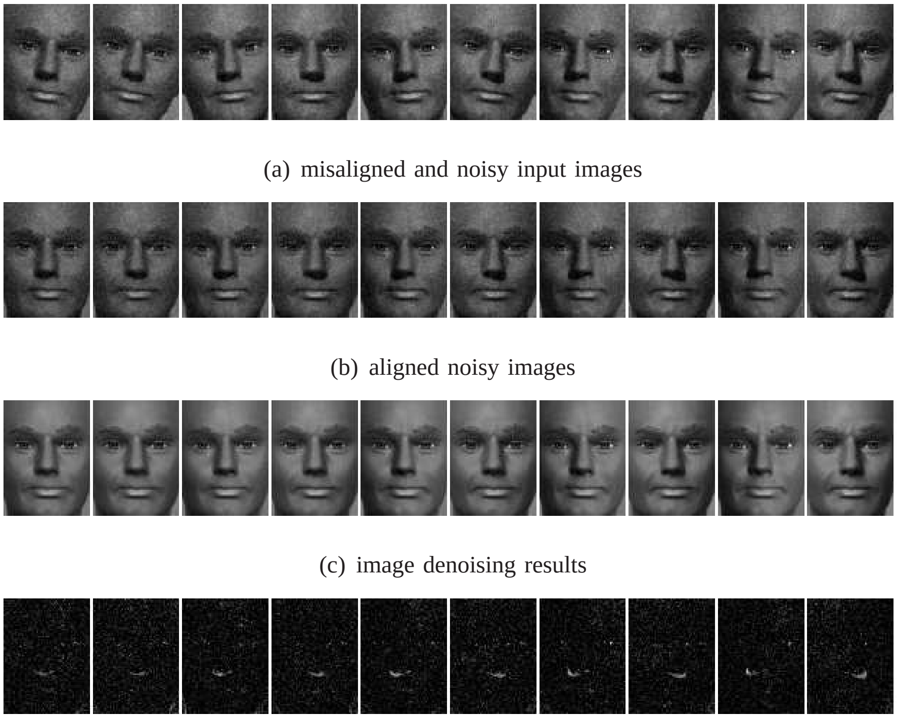


Fig. 6. **Effect of amount of occlusion coupled with misalignment.** Fraction of successful alignments for varying levels of misalignment. Translations are given as a fraction of the distance between the eyes (here, 50 pixels), while the percentage of occluded pixels reported is the average per image.

We once again use the 100 images of the dummy head for this experiment. We synthetically add occlusion to each image in the form of a square black patch at a randomly chosen location. Figure 6 shows the percentage of successful alignments by RASL for different choices of misalignment (translation along the y direction) and average percentage of occluded pixels in each input image. We observe that RASL can effectively align the images even when upto 15% of the pixels are occluded and the images are misaligned with respect to each other by upto 5 pixels along the y direction.

4) Multiple image denoising: We now demonstrate RASL as a tool to simultaneously align and denoise multiple images of the same scene. Unlike occlusions that occur as contiguous blocks in the images, here we consider corruptions that are distributed more evenly throughout the image. In particular, we consider errors that are distributed according to the random signs and support model described in [16]. According to this model, each pixel is corrupted independently with probability $\rho \in (0,1)$ and the sign of the non-zero error is uniformly distributed in $\{+1,-1\}$.



(a) misaligned and noisy input images

(b) aligned noisy images

(c) image denoising results

(d) errors

Fig. 7. **Multiple images denoising.** (a) misaligned original images with large sparse noise; (b) alignment results using RASL; (c) denoising results; (d) magnitude of the recovered errors. The images are cropped to a size of 80×60 pixels.

In this experiment, we use the 100 dummy head images described earlier. We corrupt approximately 20% of the pixels in each image (*i.e.,* $\rho = 0.2$). The results are shown in Figure 7. We observe that the output images are well-aligned with respect to each other and free of corruptions. Recently, [38] proposed an image denoising algorithm based on low-rank matrix completion. Our method differs from that work in three main aspects. Firstly, we denoise the images globally instead of in a patch-based fashion. Secondly, we do not require any information about the locations of the corrupted pixels. Thirdly, RASL recovers the global domain transformation while denoising the image simultaneously.

5) Comparison with [9]: We next perform a qualitative and quantitative comparison with the two methods proposed in [9].[9] While that work also minimizes a rank surrogate, it lacks robustness to corruption and occlusion. For compatibility with [9], we choose the canonical frame to be 49×49 pixels.[10] To each image, we apply a random Euclidean transformation whose angle of rotation is uniformly distributed in $[-10°, 10°]$ and whose x- and y-translations are uniformly distributed in $[-3, 3]$ pixels. We also synthetically occlude a randomly chosen 12×12 patch on 30 of the 100 images, thereby corrupting roughly 6% of all pixels.

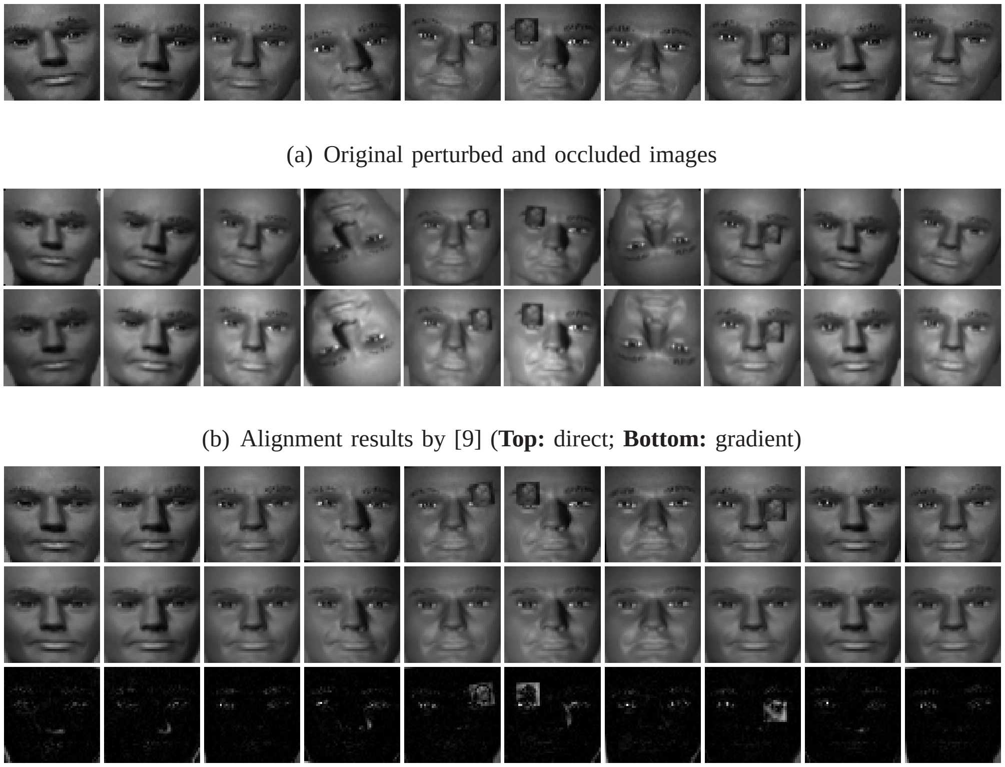
Figure 8(a) shows 10 of the 100 perturbed and occluded images. Figure 8(b) shows the alignment result using [9]. We note that eight of the 100 are flipped upside down; some of the remaining images are still obviously misaligned. Figure 8(c) shows the more visually appealing alignment produced by RASL (with $\mathbb{G}$ the similarity group $SE(2) \times \mathbb{R}_+$). We observe that RASL correctly removes the occlusions (Figure 8(c), bottom), to produce a low-rank matrix of well-aligned images (Figure 8(c), middle). The table in Figure 8(d) gives a quantitative comparison between the two algorithms.[11] Statistically, RASL produces alignments within half a pixel accuracy, with standard deviations of less than quarter of a pixel in the recovered eye corners. The performance of [9] suffers in the presence of occlusion: even with the eight flipped images excluded, the mean error is nearly two pixels.

6) Speed and scalability of RASL: The RASL formulation consists of solving a sequence of convex optimization problems. Recent advances in nuclear-norm minimization have enabled us to develop scalable algorithms for RASL. We provide the running time for an example case to give an idea of the efficiency of our algorithm. On a Macbook Pro laptop with a 2.8 GHz Intel Core 2 Duo processor and 4 GB of memory, a MATLAB implementation of RASL can align 100 images, each of size 80×60 pixels, in about 3 minutes. This is a huge improvement over the APG algorithm proposed earlier in a conference version of this work [31], which takes about 20 minutes to align the same set of images.

[9] We have actively sought implementations of other alignment methods such as TCA [12] and RPCA [13], but at the time of preparation of this paper had only received code for [9].

[10] Due to memory limitations and running time, this is the largest image size that the code of [9] can handle; as we will see in later experiments, RASL however has no problem scaling up to images of much larger sizes.

[11] We calculate all 100 images' eye corners for RASL but only the 92 un-flipped images for Vedaldi's method [9].



(a) Original perturbed and occluded images

(b) Alignment results by [9] (**Top:** direct; **Bottom:** gradient)

(c) Alignment results by RASL (**Top:** aligned input images; **Middle:** aligned images with occlusions removed; **Bottom:** recovered errors)

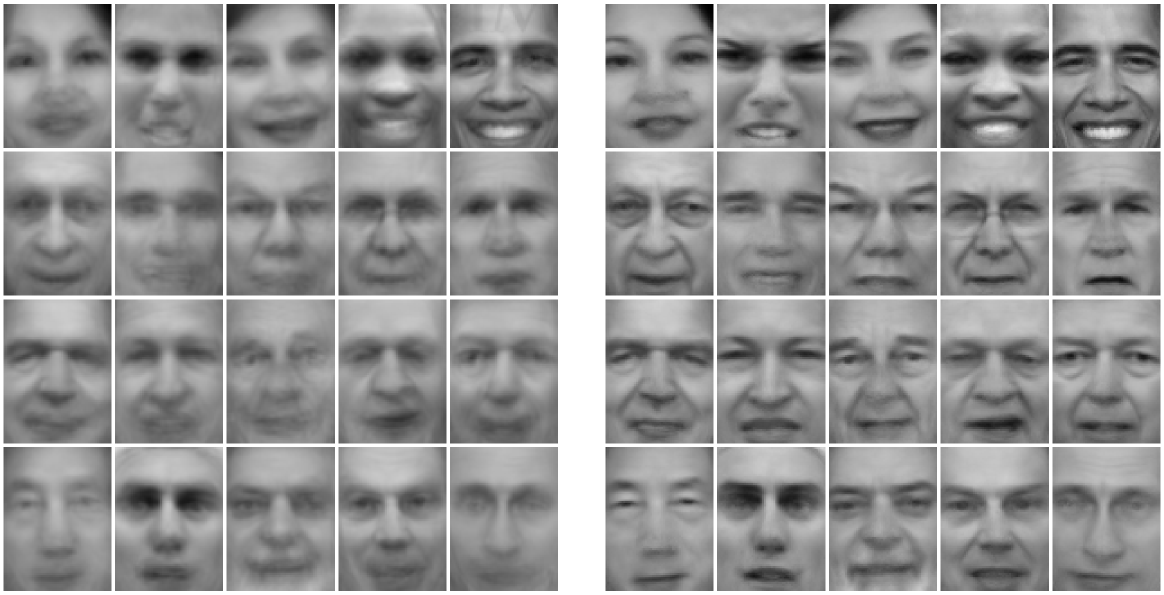
	Mean error	Error std.	Max error
Initial misalignment	2.5	1.03	4.87
[9] (direct/gradient)	1.97/1.66	1.11/0.85	5.71/4.02
RASL (this work)	**0.48**	**0.23**	**1.07**

(d) Statistics of errors in the locations of the eye corners, calculated as the distances (in pixels) from the estimated eye corners to their center.

Fig. 8. **Comparison with controlled images.** (a) 10 out of 100 images of a dummy head. (b) alignment by Vedaldi's methods [9]: *direct search* of rotation and translation (top) and *gradient descent* on a full affine transformation (bottom). (c) alignment by RASL: $D \circ \tau$ (top), low-rank approximation A (middle), and sparse errors E (bottom).

B. Qualitative Evaluation with Natural Images

1) Aligning natural face images: We next test our algorithm on more challenging images taken from the Labeled Faces in the Wild (LFW) [1] dataset of celebrity images. Unlike the controlled images in our previous example, these images exhibit significant variations in pose and facial expression, in addition to changes in illumination and occlusion.



(a) Average faces from face detector

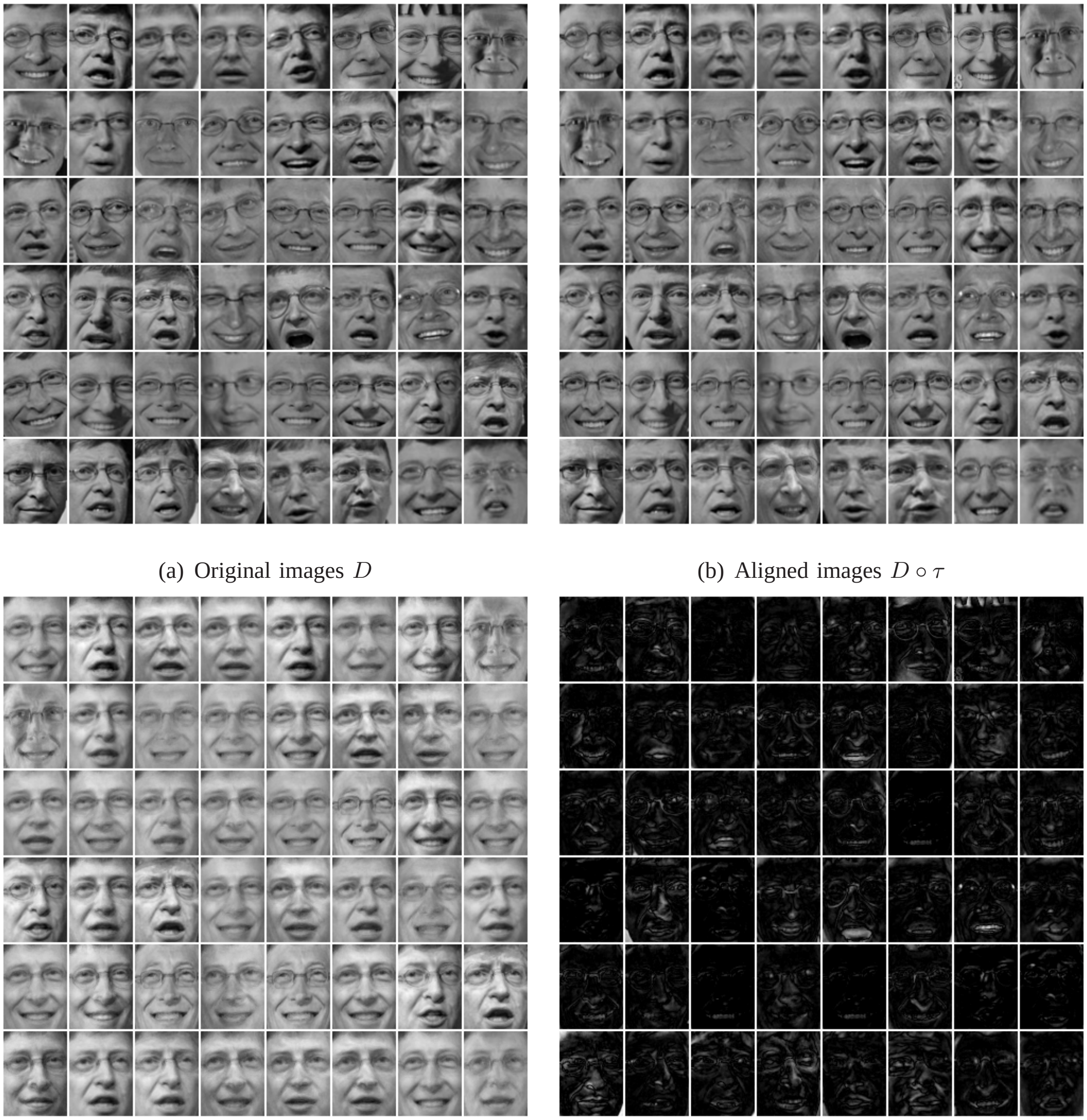
(b) Average faces after alignment

Fig. 9. **Aligning natural face images.** Average faces before and after alignment. (a) average of original images obtained using a face detector; and (b) average of the reconstructed low-rank images.

We obtain an initial estimate of the transformation in each image using the Viola-Jones face detector [39]. We again align the images to an 80×60 canonical frame. For this experiment, we use affine transformations $\mathbb{G} = \mathrm{Aff}(2)$ in RASL, to cope with the large pose variability in LFW.

Since there is no ground truth for this dataset, we verify the good performance of RASL visually by plotting the average face before and after alignment. Figure 9 shows results for some celebrities from LFW, as well as for images of Barack Obama that were separately downloaded from the Internet. We note that the average face after alignment is significantly sharper, indicating the improved alignment achieved by RASL.

As an additional example, we selected some images of Bill Gates at random from the Internet, and used RASL to align them together with a few more images from the LFW dataset. In Figure 10, we show the alignment results on all 48 images used for this experiment. The images are initially cropped by applying the face detector, as shown in Figure 10(a). We downsample the images to a 80×60 canonical frame and use the RASL algorithm with affine transformation to align the images. The alignment results are shown in Figure 10(b)-(d). We observe that large occlusions (like the Time magazine logo) and severe expression variations are effectively handled



(a) Original images D

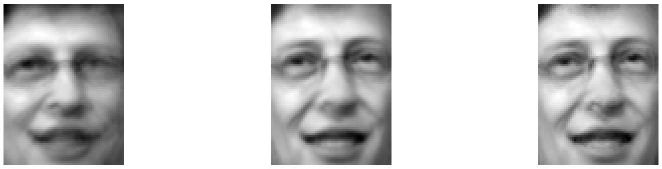
(b) Aligned images $D \circ \tau$

(c) Low-rank component A

(d) Sparse large errors E

Fig. 10. **Aligning *Bill Gates*' face images collected from the Internet.** (a) original images obtained by face detector; (b) alignment results using RASL; (c) recovered clean images; (d) recovered errors. The size of each cropped image is 80×60 pixels.

as large magnitude errors by RASL. The reason large expression changes are considered as errors is because they cannot be modeled effectively by a global transformation of the face image, as implemented in this work. We also plot the average of the face images in Figure 11 for both the input images D, the aligned images $D \circ \tau$, and the images represented by the low-rank



(a) Average $\bar{D}$ (b) Average $\bar{D} \circ \tau$ (c) Average $\bar{A}$

Fig. 11. **Qualitative evaluation from average images.** The quality of the alignment results can be assessed from the average of the 48 images of Bill Gates before and after alignment. **Left:** average input image; **Middle:** average input image after correcting for alignment; **Right:** average image after alignment and error correction. The average images after applying the recovered alignments are much sharper than the average input image.

matrix A for visual comparison. The much-sharpened average face images after alignment and error correction indicate the efficiency of the RASL algorithm. This experiment suggests that RASL could potentially be very useful for improving the performance of current face recognition systems under less-controlled or uncontrolled conditions.

2) Video stabilization: Video frames are another rich source of linearly correlated images. In this example, we demonstrate the utility of RASL for jointly aligning the frames of a video. Figure 12 shows the first 15 frames of a 140-frame video of Al Gore talking, obtained by applying a face detector to each frame independently. Due to the inherent imprecision of the detector, there is significant jitter from frame to frame. The second row shows alignment results by RASL, using affine transformations. In the third row, we show the low-rank approximation obtained after alignment, while the fourth row shows the sparse error. We note that this error compensates for localized motions such as mouth movements and eye blinking that do not fit the global motion model.

We show another example of stabilizing image frames of a video, where a portion of an iris is video-taped with a static microscopic camera. The image frames suffer from severe misalignment, which are caused by head movements, eye jitters, or dilation and contraction of the pupil, etc. The presence of noise in these images further complicates the problem. For this experiment, we use a canonical image size of 232×312 pixels. Due to the high-resolution of the images, we use a multi-scale extension of RASL to speed up the algorithm. Here, the images are progressively aligned from down-sampled versions, using the results of previous level to initialize the transformation

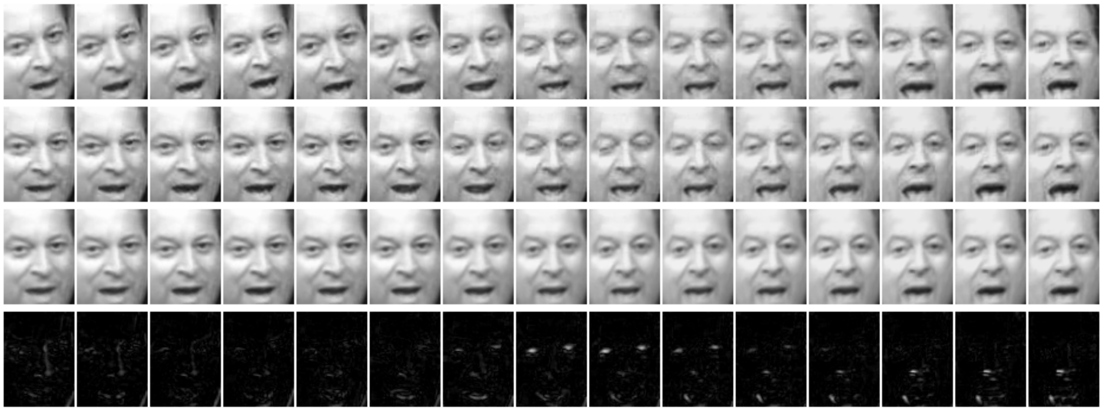


Fig. 12. **Stabilization of faces in the video. 1st row:** frames 1-15 from a 140-frame video, cropped by applying a face detector to each frame; **2nd row:** input images after alignment $D \circ \tau$; **3rd row:** recovered low rank component A; **4th row:** sparse errors E.

parameters of the next. In Figure 13, we show the result of aligning this iris video sequence, consisting of 25 frames, using RASL with an affine transformation model. We compare the original video and the aligned video with three different frames. The frames show severe jitter, blur, and intensity variation, which presents great challenges for image alignment. As can be seen from the difference images in Figure 13, the errors between any two frames is significantly reduced after alignment by RASL and become much more like random noise. Although RASL is not designed to handle additive random noise, experimental results suggest that it is stable to small amounts of noise in the image frames.

3) Aligning handwritten digits: While the previous examples concerned images and videos of human faces, RASL is a general technique capable of aligning any set of images with strong linear correlation. In this experiment, we demonstrate the applicability of our algorithm to other types of images by using it to align handwritten digits taken from the MNIST database. For this experiment, we use 100 images of the handwritten "3", of size 29×29 pixels.

Figure 14 compares the performance of RASL (using Euclidean transformation $\mathbb{G} = E(2)$) to that of [5] and [9]. RASL obtains comparably good performance on this example, despite the fact that [5] explicitly targets binary image alignment.

4) Aligning planar surfaces despite occlusions: While the previous examples used simple transformation groups such as similarity and affine, RASL can also be used with more compli-

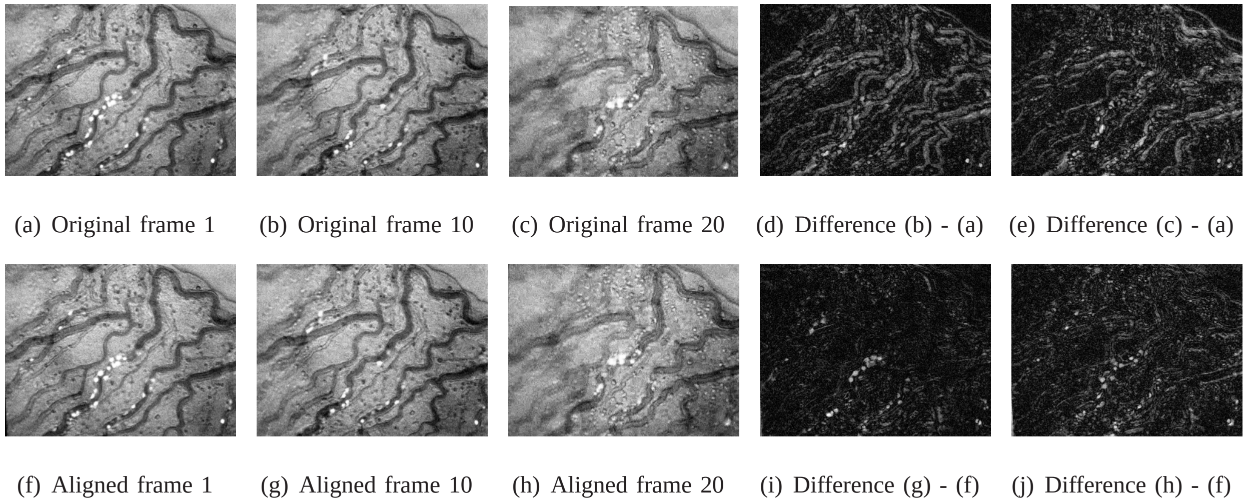
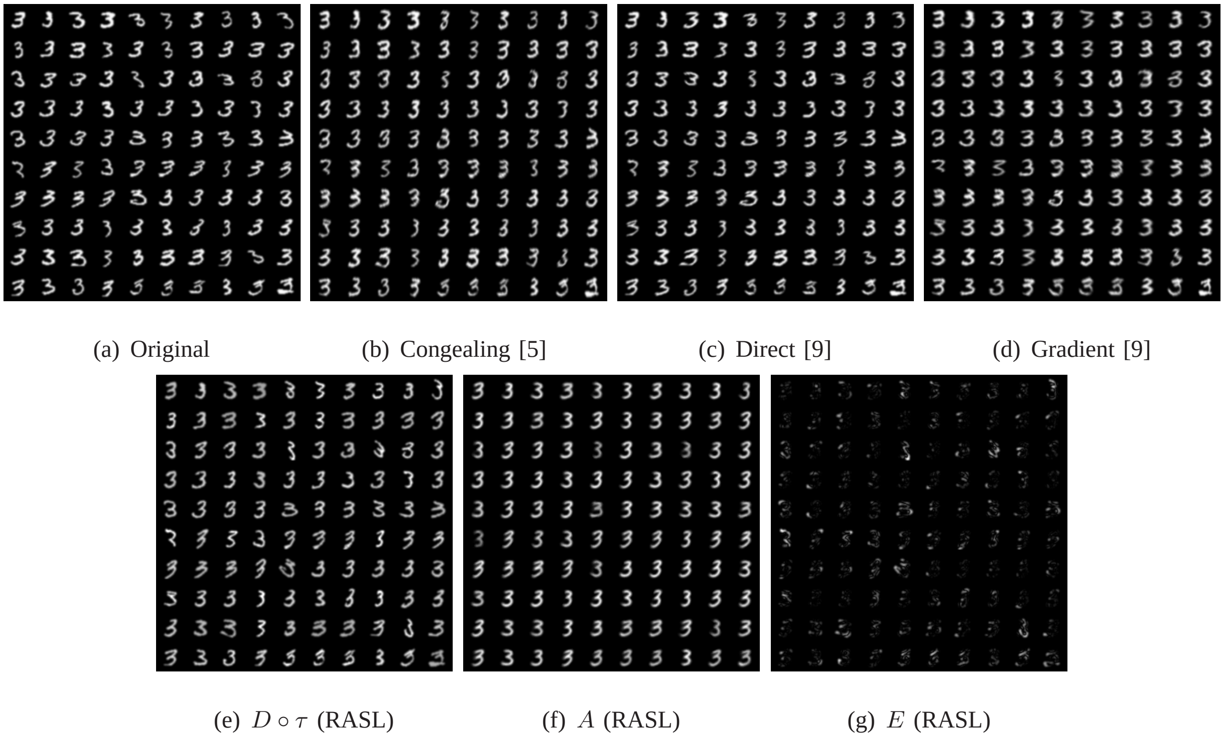


Fig. 13. **Microscopic iris video stabilization.** The original image frames 1, 10, 20 are shown in (a), (b), (c) respectively with the corresponding frames after alignment shown in (f), (g), (h); the absolute intensity difference between the frames is used to qualitatively assess the performance of RASL. The size of each cropped image is 232×312 pixels.

cated deformation models. In this example, we demonstrate how RASL can be used to align images that differ by planar homographies (i.e. $\mathbb{G} = GL(3)$). Figure 15 shows 16 images of the side of a building, taken from various viewpoints by a perspective camera, and with occlusions caused by tree branches. We manually chose three points in each image to obtain an initial affine transformation. We then used RASL, with the planar homography group of transformations, to correctly align the images to a 200×200 pixel canonical frame. As can be seen in Figure 15, RASL correctly aligns the windows and removes the branches occluding them. This example suggests that RASL could be very useful for practical tasks such as image matching, mosaicing, and inpainting.

V. CONCLUSION AND FUTURE WORK

We have presented an image alignment method that can simultaneously align multiple images by exploiting the low-rank property of aligned images. Our approach is based on recent advances in efficient matrix rank minimization that come with theoretical guarantees. The proposed algorithm consists of solving a sequence of convex optimization problems, and hence, both tractable and scalable. This allows us to simultaneously align dozens or even hundreds of images on a typical PC in matter of minutes. Furthermore, our method acts directly on the input images, and does not require any pre-filtering or feature extraction and matching. We have shown the



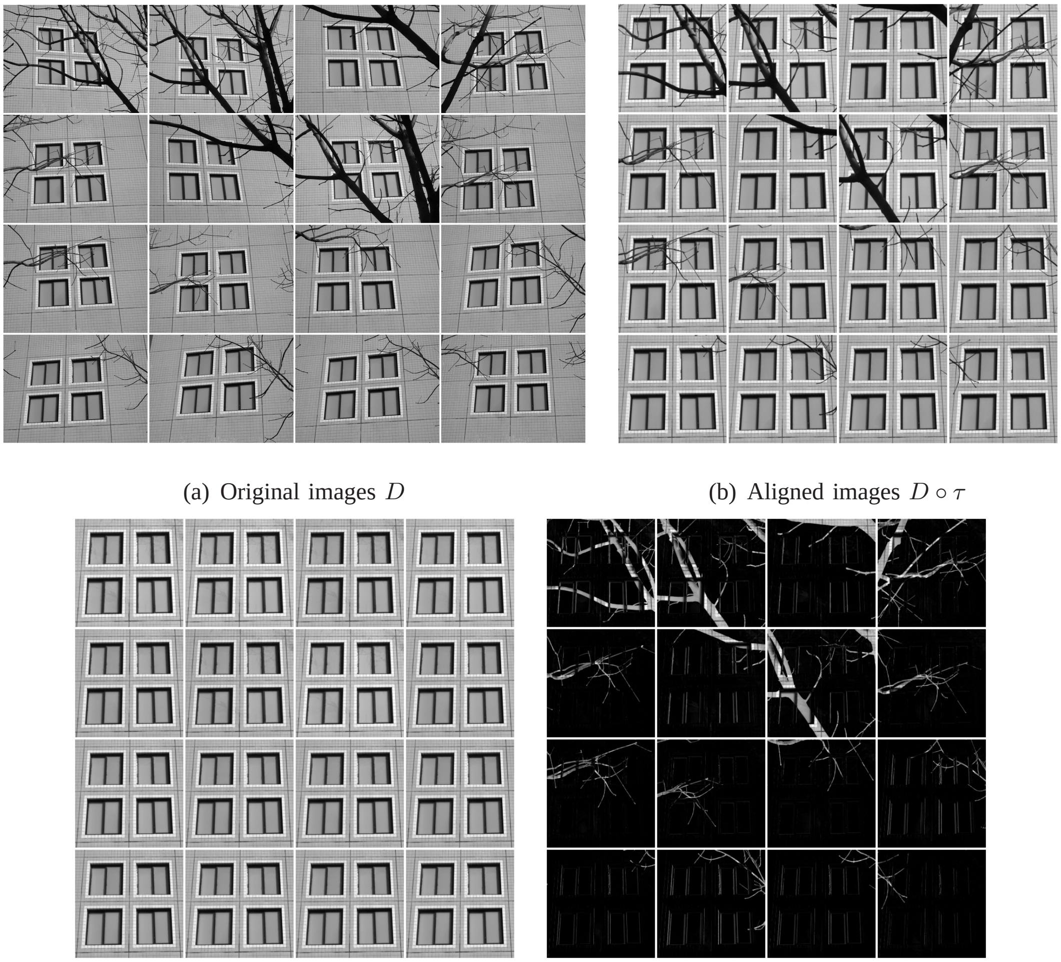
(a) Original (b) Congealing [5] (c) Direct [9] (d) Gradient [9]

(e) $D \circ \tau$ (RASL) (f) A (RASL) (g) E (RASL)

Fig. 14. **Comparison of aligning handwritten digits.** (a) original digit images; (b) aligned images using Miller's method [5]; (c) aligned images using Vedaldi's method [9] based on direct search of rotation and translation; (d) aligned images using Vedaldi's method [9]; refinement based on gradient descent on the full six parameters of the affine transformation; (e) RASL alignment result $D \circ \tau$ (f) low-rank images A (of rank 30); (g) sparse error E.

efficacy of our method with extensive experiments on images taken under laboratory conditions and on natural images of various types taken under a wide range of real-world conditions. A MATLAB implementation of our algorithm, along with sample data used in this paper, has been made publicly available for the interested reader to evaluate or use.

Currently, our method can handle one global domain transformation per image, such as affine or projective transformations. It would be useful to many practical applications if this work can be extended to handle multiple transformations in each image, where the image sequence consists of multiple independently moving objects or regions. It would also be interesting to extend this approach to the case where each input image can be deformed by wider classes of nonlinear or non-parametric domain transformations.



(a) Original images D

(b) Aligned images $D \circ \tau$

(c) Reconstructed images A

(d) Recovered occlusions E

Fig. 15. **Aligning planar homographies using RASL with ($\mathbb{G} = GL(3)$).** (a) original images from 16 views; (b) RASL alignment result $D \circ \tau$; (c) recovered low-rank component A; (d) sparse error E.

Acknowledgement

This work was partially supported by the grants NSF IIS 08-49292, NSF ECCS 07-01676, and ONR N00014-09-1-0230.

References

[1] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, “Labeled faces in the wild: A database for studying face recognition in unconstrained environments,” *Tech. Report., U. Mass. Amherst*, pp. 07–49, 2007.

[2] J. B. A. Maintz and M. A. Viergever, “A survey of medical image registration,” *Medical Image Analysis*, vol. 2, no. 1, pp. 1–36, 1998.

[3] L. G. Brown, “A survey of image registration techniques,” *ACM Computing Surveys*, vol. 24, no. 4, pp. 325–376, 1992.

[4] J. P. W. Pluim, J. B. A. Maintz, and M. A. Viergever, “Mutual-information-based registration of medical images: a survey,” *IEEE Trans. on Medical Imaging*, vol. 22, no. 8, pp. 986–1004, 2003.

[5] E. Learned-Miller, “Data driven image models through continuous joint alignment,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 236–250, 2006.

[6] G. B. Huang, V. Jain, and E. Learned-Miller, “Unsupervised joint alignment of complex images,” in *Proc. of IEEE International Conference on Computer Vision*, 2007.

[7] M. Cox, S. Lucey, S. Sridharan, and J. Cohn, “Least squares congealing for unsupervised alignment of images,” in *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition*, 2008.

[8] ——, “Least-squares congealing for large numbers of images,” in *Proc. of IEEE International Conference on Computer Vision*, 2009.

[9] A. Vedaldi, G. Guidi, and S. Soatto, “Joint alignment up to (lossy) transforamtions,” in *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition*, 2008.

[10] M. Fazel, H. Hindi, and S. Boyd, “Log-det heuristic for matrix rank minimization with applications to Hankel and Euclidean distance matrices,” in *Proc. of American Control Conference*, 2003.

[11] B. Frey and N. Jojic, “Transformed component analysis: Joint estimation of spatial transformations and image components,” in *Proc. of IEEE International Conference on Computer Vision*, 1999.

[12] ——, “Transformation-invariant clustering using the EM algorithm,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, 2003.

[13] F. de la Torre and M. Black, “Robust parameterized component analysis: Theory and applications to 2D facial appearance models,” *Computer Vision and Image Understanding*, vol. 91, no. 1-2, pp. 53–71, 2003.

[14] ——, “A framework for robust subspace learning,” *International Journal on Computer Vision*, vol. 54, no. 1-3, pp. 117–142, 2003.

[15] V. Chandrasekaran, S. Sanghavi, P. Parrilo, and A. Willsky, “Rank-sparsity incoherence for matrix decomposition,” *preprint*, 2009.

[16] E. Candès, X. Li, Y. Ma, and J. Wright, “Robust principal component analysis?” *preprint*, 2009.

[17] R. Basri and D. Jacobs, “Lambertian reflectance and linear subspaces,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, pp. 218–233, 2003.

[18] Y. Ma, S. Soatto, J. Košecká, and S. S. Sastry, *An Invitation to 3-D Vision*. Springer, 2004.

[19] J. Wright, A. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, “Robust face recognition via sparse representation,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 31, pp. 210–227, 2009.

[20] S. Baker and I. Matthews, “Lucas-Kanade 20 years on: A unifying framework,” *International Journal on Computer Vision*, vol. 56, pp. 221–255, 2004.

[21] A. Wagner, J. Wright, A. Ganesh, Z. Zhou, and Y. Ma, “Towards a practical face recognition system: Robust registration and illumination by sparse representation,” in *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition*, 2009.

[22] Z. Zhou, X. Li, J. Wright, E. Candes, and Y. Ma, “Stable principal component pursuit,” in *Proc. of International Symposium on Information Theory*, 2010.

[23] A. Lewis and S. Wright, “A proximal method for composite minimization,” *Technical Report, University of Wisconsin*, 2008.

[24] K. Jittorntrum and M. Osborne, “Strong uniqueness and second order convergence in nonlinear discrete approximation,” *Numerische Mathematik*, vol. 34, pp. 439–455, 1980.

[25] L. Cromme, “Strong uniqueness: A far-reaching criterion for the convergence analysis of iterative procedures,” *Numerishe Mathematik*, vol. 29, pp. 179–193, 1978.

[26] D. Donoho and C. Grimes, “Image manifolds which are isometric to Euclidean space,” *Journal of Mathematical Imaging and Vision (JMIV)*, vol. 23, no. 1, pp. 5–24, 2005.

[27] K. Toh and S. Yun, “An accelerated proximal gradient algorithms for nuclear norm regularized least squares problems,” *preprint*, 2009.

[28] A. Ganesh, Z. Lin, J. Wright, L. Wu, M. Chen, and Y. Ma, “Fast algorithms for recovering a corrupted low-rank matrix,” in *Proc. of Computational Advances in Multi-Sensor Adaptive Processing*, December 2009.

[29] Z. Lin, M. Chen, L. Wu, and Y. Ma, “The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices,” *UIUC Technical Report UILU-ENG-09-2215*, 2009.

[30] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific, 2004.

[31] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma, “RASL: Robust alignment via sparse and low-rank decomposition for linearly coorelated images,” in *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition*, 2010.

[32] R. Glowinski and A. Marroco, “Sur l’approximation, par elements finis d’ordre un, et la resolution, par penalisation-dualite, d’une classe de problemes de Dirichlet non lineares,” *Revuew Francaise d’Automatique, Informatique et Recherche Operationelle*, vol. 9, pp. 41–76, 1975.

[33] D. Gabay and B. Mercier, “A dual algorithm for the solution of nonlinear variational problems via finite element approximations,” *Computers and Mathematics with Applications*, vol. 2, pp. 17–40, 1976.

[34] J. Eckstein and D. Bertsekas, “On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators,” *Mathematical Programming*, vol. 55, pp. 293–318, 1992.

[35] X. Yuan and M. Tao, “Recovering low-rank and sparse components of matrices from incomplete and noisy observations,” *preprint, http://www.optimization-online.org/DB_FILE/2009/12/2510.pdf*, 2010.

[36] B. He, “Parallel splitting augmented Lagrangian methods for monotone structured variational inequalities,” *Computational Optimzation and Applications*, vol. 42, no. 2, 2009.

[37] R. Gross, I. Mathews, J. Cohn, T. Kanade, and S. Baker, “Multi-PIE,” in *Proc. of IEEE International Conference on Automatic Face and Gesture Recognition*, 2008.

[38] H. Ji, C. Liu, Z. Shen, and Y. Xu, “Robust video denoising using low rank matrix completion,” in *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition*, 2010.

[39] P. Viola and M. J. Jones, “Robust real-time face detection,” *International Journal on Computer Vision*, vol. 57, pp. 137 – 154, 2004.