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

Yigang Peng[1], Arvind Ganesh[2], John Wright[3], Wenli Xu[1] and Yi Ma[2,3]
[1]TNLIST and Dept. of Automation, Tsinghua University
[2]Dept. of Electrical and Computer Engg., University of Illinois at Urbana-Champaign
[3]Visual Computing Group, Microsoft Research Asia
pyg07@mails.tsinghua.edu.cn, abalasu2@illinois.edu, jowrig@microsoft.com,
xuwl@tsinghua.edu.cn, mayi@microsoft.com

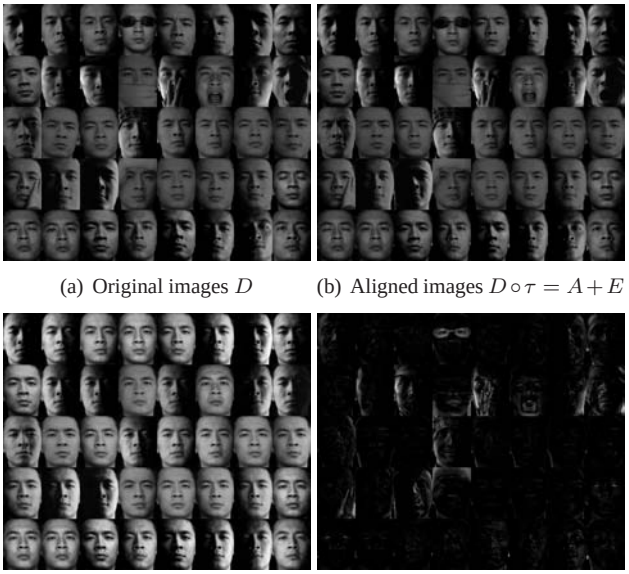
Abstract

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 with guaranteed fast convergence. We verify the efficacy of the proposed robust alignment algorithm with extensive experiments with both controlled and uncontrolled real data, demonstrating higher accuracy and efficiency than existing methods over a wide range of realistic misalignments and corruptions.

1. 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 [15]. 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 task [18, 3]: *given many images of an object or objects of interest, align them to a fixed canonical template*.

To a large extent, progress in batch image alignment has been driven by the introduction of increasingly sophisticated measures of image similarity [21]. Learned-Miller's influential *congealing* algorithm seeks an alignment that minimizes the sum of entropies of pixel values at each pixel in the batch of aligned images [16, 14]. If we stack the aligned images as the columns of a large matrix, this criterion demands that the *rows* of this matrix be nearly constant.



(a) Original images D (b) Aligned images $D \circ \tau = A + E$

(c) Low-rank component A (d) Sparse large errors E

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

Figure 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 $D \circ \tau$ in (b) can be decomposed as the sum of images from a low-rank approximation A in (c) and sparse large errors E in (d). The much sharpened average face images shown in (e) indicate the efficacy of our alignment algorithm.

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

978-1-4244-6983-3/10/$26.00 ©2010 IEEE

Conversely, the *least squares congealing* procedure of [6, 7] 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 each of the images (as those in Figure 1), the 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 [23], 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.

One 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 data (e.g. shadows, hats, glasses in Figure 1). The *Robust Parameterized Component Analysis* (RPCA) algorithm of [9] 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 [8], a problem that until recently lacked a polynomial-time algorithm with strong performance guarantees. Recent advances in rank minimization [5, 4] 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 of 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 up to one hundred images in several 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. We will publicly release the Matlab code package that has been submitted as part of the supplementary material, as well as all data used in this paper.

Organization of the paper. The remainder of the paper is organized as follows. Section 2 formulates batch image alignment as a rank minimization problem, subject to sparse corruption. Section 3 gives an efficient and effective practical solution to this problem, using convex programming. Section 4 presents experimental results demonstrating the efficacy of the proposed algorithm as well as its advantages over existing alternatives. Finally, Section 5 concludes with a discussion of several promising directions for future work.

2. Aligning Corrupted Linearly Correlated Images by Matrix Rank Minimization

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. 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 from an image and stacks it as a vector, then as a matrix

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

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 [1]. Being able to correctly identify this low-dimensional structure is crucial for many vision tasks such as face recognition.

Modeling corruption. In practice, however, this low-rank structure can be easily violated when the object of interest is partially occluded or the image is corrupted. For example, in Figure 1, the shadow, sunglasses, hat, and scarves break the linear structure of the set of face images. Thus, rather than directly observing linearly correlated images $I_1^0, \ldots, I_n^0$, it is more practical to assume that we observe $I_1 = I_1^0 + \boldsymbol{e}_1, \ldots, I_n = I_n^0 + \boldsymbol{e}_n$, where $\boldsymbol{e}_i$ is an additive error modeling the effect of such occlusions on image i. The errors $\boldsymbol{e}_i$ are large in magnitude, but affect only a fraction of the image pixels, and hence are *sparse*: most of the entries are zero. In terms of the dataset as a whole, this observation model can be written as

$$D \doteq [\mathrm{vec}(I_1) \mid \cdots \mid \mathrm{vec}(I_n)] = A + E, \qquad (2)$$

where $A = \left[\mathrm{vec}(I_1^0) \mid \cdots \mid \mathrm{vec}(I_n^0)\right] \in \mathbb{R}^{m \times n}$ is a low-rank matrix that models the common linear structure in the batch of images, and $E = [\mathrm{vec}(\boldsymbol{e}_1) \mid \cdots \mid \mathrm{vec}(\boldsymbol{e}_n)] \in \mathbb{R}^{m \times n}$ is a matrix of large-but-sparse errors that models corruption, occlusion, shadows, and specularities etc.

Modeling misalignment. The above model (correlated images with sparse errors) arises in a wide variety of applications, and has been used with great success in automatic face recognition [25]. However, it depends critically on the assumption that the given images I_i^0 are pixelwise aligned. Even small *misalignment* breaks the linear structure in the data, so that even if we could correct the errors E, the resulting matrix A would still be full rank. We

can model practical misalignments as certain transformations $\tau_1^{-1}, \ldots, \tau_n^{-1} \in \mathbb{G}$ acting on the two-dimensional domain of the images $I_1^0, \ldots, I_n^0$, respectively. In this paper, we assume that $\mathbb{G}$ is a finite-dimensional group 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 $GL(3)$. Instead of observing the original images I_i^0, we observe misaligned images $I_i^0 \circ \tau_i^{-1}$.

When both corruption and misalignment are present, in order to correctly recover the common low-rank structure in the batch of images, we must *simultaneously align the images and correct any errors in them*. This compounded problem can be formalized as follows:

Problem 1 (Robust Alignment of Correlated Images). *Let $I_1^0, \ldots, I_n^0$ be a set of linearly correlated images. Given an observation consisting of corrupted and misaligned versions $I_1 = (I_1^0 + \boldsymbol{e}_1) \circ \tau_1^{-1}, \ldots, I_n = (I_n^0 + \boldsymbol{e}_n) \circ \tau_n^{-1}$, recover the images $\{I_i^0\}$ and transformations $\{\tau_i\}$.*[1]

Our approach to solving this problem is conceptually very simple. 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 compensated for. Formally, writing $D \circ \tau$ as shorthand for $[\mathrm{vec}(I_1 \circ \tau_1) \mid \cdots \mid \mathrm{vec}(I_n \circ \tau_n)] \in \mathbb{R}^{m \times n}$:

$$\min_{A,E,\tau} \ \mathrm{rank}(A) \quad \text{s.t.} \quad D \circ \tau = A + E, \ \|E\|_0 \leq k. \quad (3)$$

Here, the ℓ^0-"norm" $\|\cdot\|_0$ counts the number of nonzero entries in the error matrix E. As we will see, it is more convenient to consider the Lagrangian form of this problem:

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

Here, $\gamma > 0$ 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).

While (4) follows naturally from our problem formulation, this optimization problem is not directly tractable: both rank and ℓ^0-norm are *nonconvex and discontinuous*, and the equality constraint $D \circ \tau = A + E$ is highly *nonlinear*. In the next section, we give an effective practical solution to this problem, building on recent advances in algorithms for robust matrix rank minimization [4, 5, 17].

3. Solution via Iterative Convex Programming

In this section, we present a practical solution to the RASL problem (4), that works quite effectively as long as the misalignments are not too large. We first relax the highly nonconvex objective function in (4) to its convex surrogate (Section 3.1). We then linearize the nonlinear equality constraint in (4) (Section 3.2), yielding a sequence of convex programs that can be solved efficiently via modern first-order optimization techniques (Section 3.3). In Section 4 we will verify the practical convergence behavior of this scheme with numerous real-data examples.

3.1. Convex relaxation

As discussed above, the optimization problem (4) is not directly tractable. One 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. 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 errors is not too large, minimizing natural convex surrogate for $\mathrm{rank}(A) + \lambda \|E\|_0$ can *exactly* recover A [4].[2] This convex relaxation replaces $\mathrm{rank}(\cdot)$ with the *nuclear norm* or sum of the singular values: $\|A\|_* \doteq \sum_{i=1}^m \sigma_i(A)$, 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 (4) yields a new optimization problem:

$$\min_{A,E,\tau} \ \|A\|_* + \lambda \|E\|_1 \quad \text{s.t.} \quad D \circ \tau = A + E. \quad (5)$$

Theoretical considerations in [4] suggest that the weighting parameter λ should be of the form $C/\sqrt{m}$ where C is a constant, typically set to be $C \approx 1$. The new objective function is non-smooth, but now continuous and convex.

3.2. Iterative linearization

The main remaining difficulty in solving (5) 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 $\Delta\tau \in \mathbb{R}^{p \times n}$, 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 $A, E, \Delta\tau$:

$$\min_{A,E,\Delta\tau} \|A\|_* + \lambda \|E\|_1 \ \text{ s.t. } \ D \circ \tau + \sum_{i=1}^n J_i \Delta\tau_i \epsilon_i^T = A + E. \quad (6)$$

[1] We here consider two solutions are equivalent if they only differ by a single common transformation acting on all the images. This ambiguity can be easily resolved in practice if we set one image as the reference.

[2] For appropriate random matrix models, the relaxation succeeds whenever $\mathrm{rank}(A) < C_1 m / \log(m)$ and $\|E\|_0 < C_2 mn$ [4] for some constants C_1, C_2. Similar guarantees can be proved for the linearized convex optimization to be introduced in Section 3.3, but are not the main focus of this paper.

Algorithm 1 (Outer loop of RASL)

INPUT: Images $I_1, \dots, I_n \in \mathbb{R}^{w \times h}$, initial transformations $\tau_1, \dots, \tau_n$ in certain parametric group $\mathbb{G}$, weight $\lambda > 0$.

WHILE not converged **DO**

step 1: compute Jacobian matrices w.r.t. transformation:

$$J_i \leftarrow \frac{\partial}{\partial \zeta}\left(\frac{\mathrm{vec}(I_i \circ \zeta)}{\|\mathrm{vec}(I_i \circ \zeta)\|_2}\right)\Big|_{\zeta=\tau_i}, \quad i = 1, \dots, n;$$

step 2: warp and normalize the images:

$$D \circ \tau \leftarrow \left[\frac{\mathrm{vec}(I_1 \circ \tau_1)}{\|\mathrm{vec}(I_1 \circ \tau_1)\|_2} \Big| \dots \Big| \frac{\mathrm{vec}(I_n \circ \tau_n)}{\|\mathrm{vec}(I_n \circ \tau_n)\|_2}\right];$$

step 3: solve the linearized convex optimization:

$$(A^*, E^*, \Delta\tau^*) \leftarrow \underset{A,E,\Delta\tau}{\arg\min} \quad \|A\|_* + \lambda\|E\|_1$$
$$\text{s.t. } D \circ \tau + \sum_{i=1}^{n} J_i \Delta\tau_i \epsilon_i^T = A + E;$$

step 4: update transformations: $\tau \leftarrow \tau + \Delta\tau^*$;

END WHILE

OUTPUT: solution A^*, E^*, τ to problem (5).

Algorithm 2 (Linearized Inner Loop of RASL)

INPUT: $\mu^0 > 0$, $(A^0, E^0, \Delta\tau^0) \in \mathbb{R}^{m\times n} \times \mathbb{R}^{m\times n} \times \mathbb{R}^{p\times n}$. Set $t^1 = t^0 = 1$, $(A^1, E^1, \Delta\tau^1) = (A^0, E^0, \Delta\tau^0)$, $k = 1$.

WHILE not converged **DO**

step 1: compute proximal points:

$Y_A^k = A^k + \frac{t^{k-1}-1}{t^k}(A^k - A^{k-1})$,

$Y_E^k = E^k + \frac{t^{k-1}-1}{t^k}(E^k - E^{k-1})$,

$Y_{\Delta\tau}^k = \Delta\tau^k + \frac{t^{k-1}-1}{t^k}(\Delta\tau^k - \Delta\tau^{k-1})$;

step 2: gradient step:

$G^k = Y_A^k + Y_E^k - D \circ \tau - \sum_{i=1}^n J_i Y_{\Delta\tau}^k \epsilon_i \epsilon_i^T$,

$G_A^k = Y_A^k - \rho^{-1} G^k, \quad G_E^k = Y_E^k - \rho^{-1} G^k$,

$G_{\Delta\tau}^k = Y_{\Delta\tau}^k + \rho^{-1} \sum_{j=1}^n J_j^T G^k \epsilon_j \epsilon_j^T$;

step 3: soft-thresholding:

compute the reduced SVD (U, S, V) of G_A^k,

$A^{k+1} = U\mathcal{T}_{\mu^k/\rho}(S)V^T, \quad E^{k+1} = \mathcal{T}_{\lambda\mu^k/\rho}(G_E^k)$,

$\Delta\tau^{k+1} = G_{\Delta\tau}^k$;

step 4: update:

$t^{k+1} = \frac{1}{2} + \frac{1}{2}\sqrt{1 + 4(t^k)^2}, \quad \mu^{k+1} = \max\{0.9\mu^k, \bar{\mu}\}$;

END WHILE

OUTPUT: solution $(A, E, \Delta\tau)$ to problem (7), and hence (6).

Because the linearization only holds locally, we should not expect the solution $\tau + \Delta\tau$ from (6) to exactly solve (5). To find the (local) minimum of (5), we repeatedly linearize about our current estimate of τ and solve a sequence of convex programs of the form (6).[3] As we will see in Section 4, 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.

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.

3.3. Fast algorithm for the linearized inner loop

The main computational cost in Algorithm 1 comes in solving the linearized convex optimization problem (6) at each iteration. 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 [19, 20, 2, 22, 17]. In this section, we show how one such fast first-order method, the Accelerated Proximal Gradient (APG) algorithm [2, 22, 17], can be adapted to efficiently solve (6).

The APG approach replaces the equality constraint in (6) with a penalty function $f(A, E, \Delta\tau) \doteq \frac{1}{2}\big\|A + E - D \circ \tau - \sum_{i=1}^n J_i \Delta\tau_i \epsilon_i^T\big\|_F^2$, and instead solves the unconstrained optimization

$$\min_{A,E,\Delta\tau} \|A\|_* + \lambda\|E\|_1 + \tfrac{1}{\mu} f(A, E, \Delta\tau). \tag{7}$$

As $\mu \searrow 0$, the optimal solution to (7) approaches the optimal solution set of (6).

The algorithm solves (7) by forming separable quadratic approximations to the data fidelity term $f(A, E, \Delta\tau)$ at a special sequence of points $Y^k = (Y_A^k, Y_E^k, Y_{\Delta\tau}^k)$, conspicuously chosen to achieve essentially an optimal convergence rate for first-order methods [19, 2].[4] At each step k, the next iterate $(A^{k+1}, E^{k+1}, \Delta\tau^{k+1})$ is obtained as the solution to

$$\min_{A,E,\Delta\tau} \|A\|_* + \lambda\|E\|_1 + \tfrac{\rho}{2\mu}\|(A, E, \Delta\tau) - (G_A^k, G_E^k, G_{\Delta\tau}^k)\|_F^2. \tag{8}$$

Here, $\rho > 0$ will be specified below, and $G^k = (G_A^k, G_E^k, G_{\Delta\tau}^k) = Y^k - \rho^{-1}\nabla f|_{Y^k}$. Because the quadratic term in (8) is separable, (8) can be solved very efficiently via the soft-thresholding operator:

$$\mathcal{T}_\xi(x) = \begin{cases} \mathrm{sign}(x)\,(|x| - \xi), & |x| > \xi, \\ 0, & |x| \le \xi. \end{cases} \tag{9}$$

The iterate E^{k+1} is given by soft-threholding the entries of G_E^k, while A^{k+1} is given by soft-thresholding the singular values of G_A^k. We summarize this complete procedure as Algorithm 2, whose global optimality and convergence rate is guaranteed by Proposition 1.

Proposition 1 (Global Convergence of the Inner Loop). *Let L_f denote the Lipschitz constant of ∇f, which satisfies*

[3] This kind of iterative linearization has a long history in gradient algorithms for batch image alignment (see [23] and references therein). More recently a similar iterative convex programming approach was proposed for single-to-batch image alignment in face recognition [24].

[4] Space precludes a more detailed discussion of the choice of Y_k; we refer the interested reader to [19, 2, 17].

$$L_f \;\leq\; \sqrt{3(2+\omega^2)\max\{1,\omega^2\}}, \qquad (10)$$

where $\omega \doteq \max_i \{\|J_i\|_{2,2}\}$. *Then if* $\rho \geq L_f$, *the sequence* $(A^k, E^k, \Delta\tau^k)$ *generated by Algorithm 2 converges to the global optima of (7) with a non-asymptotic convergence rate of* $O(1/k^2)$.

Proof. The derivation of the Lipschitz constant of ∇f is given in the supplemental materials. The proof of convergence follows from more general results in [2, 22]. □

In our experiments, we always set ρ to be equal to the righthand side of (10).

3.4. Implementation details for Algorithm 2

(i). Stopping criterion. As suggested in [22], we terminate the iteration when a particular subgradient of the cost function in (7),

$$S^k \doteq \rho\left((Y_A^{k-1}, Y_E^{k-1}, Y_{\Delta\tau}^{k-1}) - (A^k, E^k, \Delta\tau^k)\right) + \nabla f(A^k, E^k, \Delta\tau^k) - \nabla f(Y_A^{k-1}, Y_E^{k-1}, Y_{\Delta\tau}^{k-1})$$

is sufficiently small in magnitude. In practice, the inner loop is terminated when

$$\frac{\|S^k\|_F}{\rho\max\left\{1, (\|A^k\|_F^2 + \|E^k\|_F^2 + \|\Delta\tau^k\|_F^2)^{1/2}\right\}} \leq \varepsilon, \quad (11)$$

where $\varepsilon > 0$ is a predefined tolerance. For our experiments, we set $\varepsilon = 10^{-6}$.

(ii). Improving convergence via QR decomposition of Jacobian matrix. The Lipschitz constant of the gradient ∇f, denoted L_f, affects the speed of convergence of the algorithm. In particular, the larger the L_f, the slower the convergence. It can be seen from equation (10) that L_f depends directly on the Jacobian matrices. Hence, it is desirable to have Jacobian matrices with small spectral norms.

One possible strategy to achieving faster convergence is to compute the QR factorization of the Jacobian matrices $J_i = Q_i R_i^T$, and solve (7) using Algorithm 2 with the orthogonal Q_i's, in place of the J_i's. The output of the algorithm in this case would be $\Delta\tau_i' = R_i\Delta\tau_i$. This procedure keeps the value of L_f small, and the original $\Delta\tau_i$ can be retrieved easily from $\Delta\tau_i'$.

(iii). Fast continuation techniques. As mentioned earlier, the solution to (7) approaches the optimal solution set of (6) as μ approaches zero. It has been suggested in [13, 22, 17] that employing a continuation technique on μ can yield drastically faster convergence when compared to using a fixed μ. The continuation is carried out by monotonically decreasing the value of the relaxation parameter μ every iteration, until it reaches a pre-defined lower bound $\bar{\mu} > 0$, beyond which it is held constant. Although the theoretical convergence rate is still $O(k^{-2})$, in practice continuation significantly reduces the number of iterations needed to converge. For all our experiments, we set $\mu^0 = \|D\|_{2,2}$, and $\mu^k = \max\{0.9\mu^{k-1}, \bar{\mu}\}$, where $\bar{\mu} = 10^{-4}\,\mu^0$.

4. Experiments

In this section, we demonstrate the efficacy of RASL on a variety of alignment tasks. Unless otherwise stated, we always set $\lambda = 1/\sqrt{m}$ in the RASL algorithm. We first quantitatively verify the correctness of our algorithm on a controlled example, 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 [15]. Experiments on video data and handwritten digits further demonstrate the generality and broad applicability of our method. Finally, an example of aligning perspective images of a planar surface demonstrates its ability to cope with more complicated deformations such as planar homographies.

(i). Quantitative validation with controlled images. We verify the correctness of the algorithm using 100 images of a dummy head taken under varying illumination (see Figure 3 top for an example). Because the relative position between the camera and the dummy is fixed, the ground truth alignment is known.

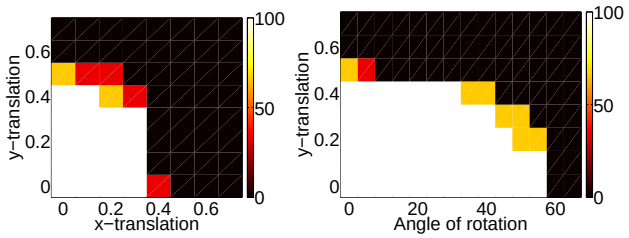
Large region of attraction for RASL. We examine RASL's ability to cope with varying levels of misalignment. The task is to align the images to an 80×60 pixel canonical frame, in which the distance between the outer eye corners is normalized to 50 pixels[5]. We synthetically perturb each of the input 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 2(a) shows the fraction of successes over three 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 2(b), we fix $x_0 = 0$ and plot the fraction of successes for 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°!

Comparison with [23]. We next perform a qualitative and quantitative comparison with the two methods given in [23].[6] While that work also minimizes a rank surrogate, it lacks robustness to corruption and occlusion. For compatibility with [23], we choose the canonical frame to be

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

[6] We have actively sought implementations of other alignment methods such as TCA [12] and RPCA [9], but at the time of preparation of this paper had only received code for [23].



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

Figure 2. **Large region of attraction for RASL.** 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 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°.

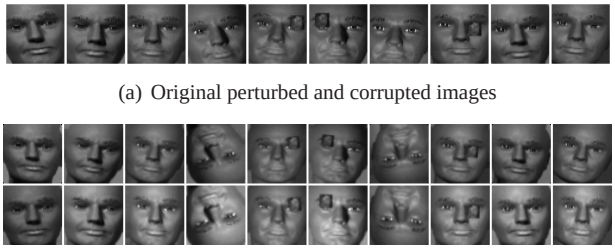
49×49 pixels.[7] To each image, we apply a random Euclidean transformation whose angle of rotation is uniformly distributed in $[-10^\circ, 10^\circ]$ and whose x- and y-translations are uniformly distributed in $[-3, 3]$ pixels. We also synthetically occlude a randomly chosen 12×12 patch on 30 of the 100 images, corresponding to roughly 6% pixels corrupted.

Figure 3(a) shows 10 of the 100 perturbed and occluded images. Figure 3(b) shows the alignment result using [23]. Eight of the 100 are flipped upside down; some of the remaining images are still obviously misaligned. Figure 3(c) shows the more visually appealing alignment produced by RASL (with $\mathbb{G}$ the similarity group $SE(2) \times \mathbb{R}_+$). Notice that RASL correctly removes the occlusions (Figure 3(c) bottom), to produce a rank 48 matrix of well-aligned images (Figure 3(c), middle). The table in Figure 3(d) gives a quantitative comparison between the two algorithms.[8] Statistically, RASL produces alignments within half a pixel accuracy, with standard deviations of less than quarter a pixel in the recovered eye corners. The performance of [23] suffers in the presence of occlusion: even with the eight flipped images excluded, the mean error is nearly two pixels.

Speed and scalability of RASL. For this example, on a 2.8 GHz Intel Pentium 4 machine with 1.5 GB RAM, our Matlab implementation of RASL requires less than 24 minutes to align the 100 images of size 49×49, whereas [23] requires over 13 hours. Later examples will show RASL works with much larger images. This impressive computational efficiency is a direct result of using appropriate convex optimization tools for rank minimization.

[7] Due to memory limit and running time, this is the largest image size that the code of [23] can handle; as we will see in later experiments, RASL however has no problem scaling up to images of much larger sizes.

[8] We calculate all 100 images' eye corners for RASL but only the 92 un-flipped images for Vedaldi's [23].



(a) Original perturbed and corrupted images

(b) Alignment results by [23] (Top: direct; Bottom: gradient)

(c) Alignment results by RASL

	Mean error	Error std.	Max error
Initial misalignment	2.5	1.03	4.87
[23] (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 eye corners, calculated as the distances from the estimated eye corners to their center. Errors in each eye and in individual coordinates are reported in the supplementary material.

Figure 3. **Comparison with controlled images.** (a) 10 out of 100 images of a dummy head. (b) alignment by Vedaldi's methods [23]: *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).

(ii). Aligning natural face images. We next test our algorithm on more challenging images taken from the Labeled Faces in the Wild (LFW) [15] 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 illumination and occlusion.

We obtain an initial estimate of the transformation in each image using an off-the-shelf face detector. We again align the images to an 80×60 canonical frame. For this experiment we use affine transformations $\mathbb{G} = \text{Aff}(2)$ 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 4 shows results for 15 celebrities from LFW, as well as Barack Obama whose images were separately downloaded from the Internet. Notice that the average face after alignment is significantly clearer, indicating the improved alignment achieved by RASL. The supplementary material contains additional examples from this dataset, showing the low-rank approximation obtained by RASL, and demonstrating its ability to correct errors in those real images. This result suggests that RASL could potentially be very useful for improving the performance of current face recognition systems under less-controlled or uncontrolled conditions.

(a) Average faces from face detector (b) Average faces after alignment

Figure 4. **Aligning natural face images.** Average faces before and after alignment. (a) average of original images from a face detector; and (b) average of the reconstructed low-rank images.

Figure 5. **Stabilization of faces in the video. 1st row:** frames 1-15 from a 140-frame video, aligned by applying a face detector to each frame; **2nd row:** RASL alignment result $D \circ \tau$; **3rd row:** recovered images A of rank 64; **4th row:** sparse error E.

(iii). Stabilization of faces in video. 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 5 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. Notice that this error compensates for localized motions such as mouth movements and eye blinking that do not fit the global motion model. We encourage the reader to refer to the supplementary material to see the entire video sequence – the recovered low-rank component even automatically repairs certain video compression artifacts. These results suggest the potential of RASL as a general tool for video stabilization, compression, and object tracking.

(iv). 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 6 compares the performance of RASL (using Euclidean transformation $\mathbb{G} = E(2)$) to that of [16] and [23]. RASL obtains comparably good performance on this example, despite the fact that [16] explicitly targets binary image alignment.

(v). 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 complicated 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 7 shows 16 images of the side of a building, taken from various viewpoints by a perspective camera, and with various occlusions due to tree branches. We manually choose three points on the image and obtain an initial affine transformation estimation for each image to initialize the transformation, and then use RASL together with a homography transformation to correctly align them to a 200×200 pixel canonical frame. As we can see from Figure 7, 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.

5. Discussion and Future Work

One of the most important questions for future work is how to effectively extend the RASL framework to handle more general classes of transformation groups such as nonrigid and nonparametric. From a theoretical standpoint, it would also be desirable to give guarantees for the amount of misalignment or corruption the algorithm can handle. The experiments in Section 4 demonstrate the surprising effectiveness and efficiency of RASL for batch image alignment, and immediately suggest applications in automatic face recognition, video stabilization and tracking, and image mosaicing, inpainting, super-resolution etc. Further customizing it to best meet the needs of such specific application scenarios and other type of signals (e.g. speech, bioinformatic data) is an important direction for future work. A MATLAB implementation of our algorithm is available at `http://perception.csl.illinois.edu/matrix-rank/rasl.html`.

References

[1] R. Basri and D. Jacobs. Lambertian reflectance and linear subspaces. *PAMI*, 25:218–233, 2003.

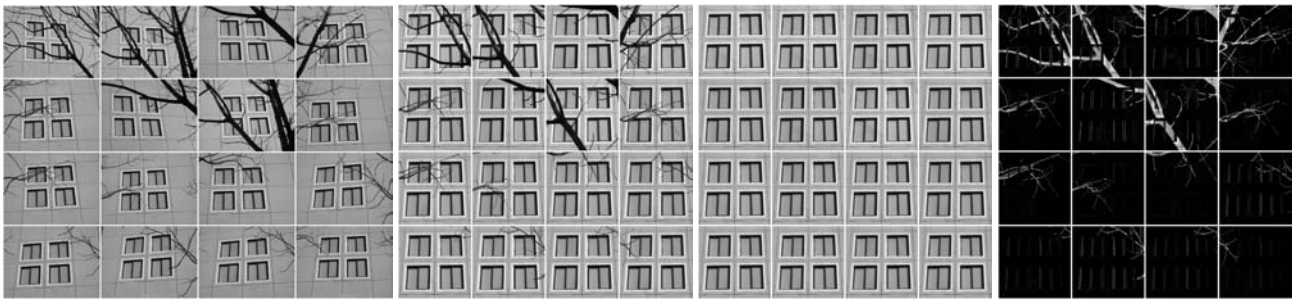
[2] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problem. *SIAM Journal on Imaging Sciences*, pages 183–202, 2008.

[3] L. G. Brown. A survey of image registration techniques. *ACM Computing Surveys*, 24(4):325–376, 1992.

[4] E. Candes, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *preprint*, 2009.

(a) Original (b) Congealing [16] (c) Direct [23] (d) Gradient [23] (e) $D \circ \tau$ (RASL) (f) A (RASL) (g) E (RASL)

Figure 6. **Comparison of aligning handwritten digits.** (a) original digit images; (b) aligned images using Miller's method [16]; (c) aligned images using Vedaldi's method [23] based on direct search of rotation and translation; (d) aligned images using Vedaldi's method [23]; 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.



(a) Original homography images (b) Aligned images $D \circ \tau$ (c) Reconstructed images A (d) Removed occlusions E

Figure 7. **Aligning planar homographies using RASL with ($\mathbb{G} = GL(3)$).** (a) original images from 16 views; (b) RASL alignment result $D \circ \tau$; (c) reconstructed low-rank images A (of rank 7); (d) sparse error E.

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

[6] M. Cox, S. Lucey, S. Sridharan, and J. Cohn. Least squares congealing for unsupervised alignment of images. In *CVPR*, 2008.

[7] M. Cox, S. Lucey, S. Sridharan, and J. Cohn. Least-squares congealing for large numbers of image. In *ICCV*, 2009.

[8] F. de la Torre and M. Black. A framework for robust subspace learning. *IJCV*, 54(1-3):117–142, 2003.

[9] F. de la Torre and M. Black. Robust parameterized component analysis: Theory and applications to 2D facial appearance models. *CVIU*, 91(1-2):53–71, 2003.

[10] M. Fazel, H. Hindi, and S. Boyd. Log-det heuristic for matrix rank minimization with applications to Hankel and Euclidean distance matrices. In *ACC*, 2003.

[11] B. Frey and N. Jojic. Transformed component analysis: Joint estimation of spatial transformations and image components. In *ICCV*, 1999.

[12] B. Frey and N. Jojic. Transformation-invariant clustering using the EM algorithm. *PAMI*, 25, 2003.

[13] E. Hale, W. Yin, and Y. Zhang. Fixed-point continuation for l1-minimization: Methodology and convergence. *SIAM J. on Optimization*, 19:1107–1130, 2008.

[14] G. B. Huang, V. Jain, and E. Learned-Miller. Unsupervised joint alignment of complex images. In *ICCV*, 2007.

[15] 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*, pages 07–49, 2007.

[16] E. Learned-Miller. Data driven image models through continuous joint alignment. *PAMI*, 28:236–250, 2006.

[17] Z. Lin, A. Ganesh, J. Wright, L. Wu, M. Chen, and Y. Ma. Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix. *UIUC Technical Report UILU-ENG-09-2214*, 2009.

[18] J. B. A. Maintz and M. A. Viergever. A survey of medical image registration. *Medical Image Analysis*, 2(1):1–36, 1998.

[19] Y. Nesterov. A method for unconstrained convex minimization problem with the rate of comvergence o($1/k^2$). *Doklady AN USSR (translated as Soviet Math. Docl)*, 1983.

[20] Y. Nesterov. Smooth minimization of non-smooth functions. *Math. Program., Serie A*, pages 127–152, 2005.

[21] 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*, 22(8):986–1004, 2003.

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

[23] A. Vedaldi, G. Guidi, and S. Soatto. Joint alignment up to (lossy) transforamtions. In *CVPR*, 2008.

[24] 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 *CVPR*, 2009.

[25] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *PAMI*, 31:210–227, 2009.