Sparse Online Low-Rank Projection and Outlier Rejection (SOLO) for 3-D Rigid-Body Motion Registration

Chris Slaughter, Allen Y. Yang, Justin Bagwell, Costa Checkles, Luis Sentis, and Sriram Vishwanath

***Abstract*— Motivated by an emerging theory of robust low-rank matrix representation, in this paper, we introduce a novel solution for online rigid-body motion registration. The goal is to develop algorithmic techniques that enable a robust, real-time motion registration solution suitable for low-cost, portable 3-D camera devices. Assuming 3-D image features are tracked via a standard tracker, the algorithm first utilizes Robust PCA to initialize a low-rank shape representation of the rigid body. Robust PCA finds the global optimal solution of the initialization, while its complexity is comparable to singular value decomposition. In the online update stage, we propose a more efficient algorithm for sparse subspace projection to sequentially project new feature observations onto the shape subspace. The lightweight update stage guarantees the real-time performance of the solution while maintaining good registration even when the image sequence is contaminated by noise, gross data corruption, outlying features, and missing data. The state-of-the-art accuracy of the solution is validated through extensive simulation and a real-world experiment, while the system enjoys one to two orders of magnitude speed-up compared to well-established RANSAC solutions.**

I. Introduction

Rigid body motion registration (RBMR) is one of the fundamental problems in machine vision and robotics. Given a dynamic scene that contains a (dominant) rigid-body object and a cluttered background, certain salient image feature points can be extracted and tracked with considerable accuracy across multiple image frames [18]. The task of RBMR then involves identifying the image features that are associated only with the rigid-body object in the foreground and subsequently recovering its rigid-body transformation across multiple frames. Traditionally, RBMR has been mainly conducted in 2-D image space, with the assumption of the camera projection model from simple orthographic projection [21] to more realistic camera models such as paraperspective [14] and affine [10]. In problems such as RBMR, Structure from Motion (SfM), and motion segmentation [11], [24], a fundamental observation is that a data matrix that contains the coordinates of tracked image features in column form can be factorized as a camera matrix that represents the motion and a shape matrix that represents the shape of the rigid body in the world coordinates. Furthermore, if the data are noise-free, then the feature vectors in the data matrix lie in a 4-D subspace, as the rank of the shape matrix in the world coordinates is at most four [21].

In practice, the RBMR problem can become more challenging if the tracked image features are perturbed by moderate noise, gross image corruption (e.g., when the features are occluded), and missing data (e.g., when the features leave the field of view). In robust statistics, it is well known that the optimal solution to recovering a subspace model when the data is complete yet affected by Gaussian noise is *singular value decomposition* (SVD). Solving other image nuisances caused by gross measurement error corresponds to the problem of robust estimation of a low-dimensional subspace model in the presence of corruption and missing data. In [8], for instance, the issue of missing data was addressed by robustifying SVD via Power Factorization. In [4], the same issue was addressed by an iterative imputation strategy.

In the case of outlier rejection, arguably the most popular robust model estimation algorithm in computer vision is Random Sample Consensus (RANSAC) [7]. In the context of RBMR, the standard procedure of RANSAC is to apply the iterative *hypothesize-and-verify* scheme on a frame-by-frame basis to recover the rigid-body motion [22], [20], [25]. In the context of dimensionality reduction, RANSAC can also be applied to recover low-dimensional subspace models [27], such as the above shape model in motion registration. Another approach closely related to this work is known as *robust factorization* in motion registration, whereby one can use iteratively reweighted least-squares (IRLS) [1] or ℓ_1-norm [15] to update the factorization of the camera motion and object shape. In the emerging compressive sensing theory, the approach is also known as *basis-pursuit denoising* (BPDN) [5].

Nevertheless, the aforementioned solutions have two major drawbacks. In the case of missing data, methods such as Power Factorization or incremental SVD cannot guarantee the global convergence of the estimate [8], [4]. In the case of outlier rejection, the RANSAC procedure is known to be expensive to deploy in a real-time, online fashion, such as in *simultaneous localization and mapping* (SLAM) [26], [16], and the BPDN approach fails to take advantage of the low-rank condition during the factorization procedure. Therefore, a better solution should provide provable global optimality to compensate missing data, image corruption, and erroneous feature tracks, and at the same time should be more efficient

C. Slaughter, J. Bagwell, C. Checkles and S. Vishwanath are with Electrical and Computer Engineering Department, University of Texas, Austin, USA. <chris.c.slaughter@gmail.com, justindbagwell@mail.utexas.edu, ccheckles@utexas.edu, sriram@austin.utexas.edu>

A. Yang is with the EECS Department, University of California, Berkeley, USA. <yang@eecs.berkeley.edu>

L. Sentis is with the Mechanical Engineering Depatment, University of Texas, Austin, USA. <lsentis@austin.utexas.edu>

This work was supported in part by the ONR, an Intel Graduate Fellowship, NSF CNS-0905200, ARO MURI W911NF-06-1-0076, and a Willow Garage gift.

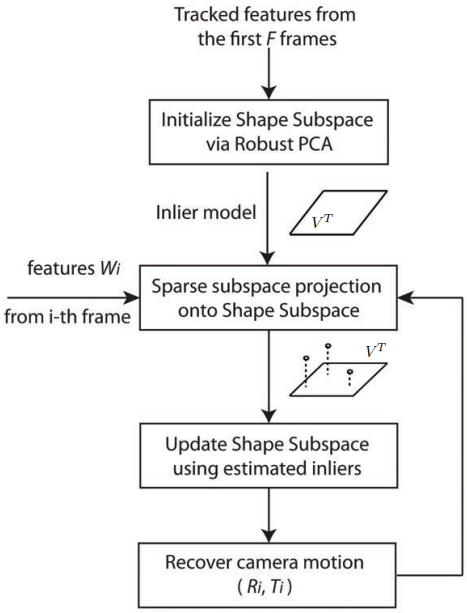


Fig. 1. Flow diagram of the SOLO algorithm.

to recover rigid-body motion from a video sequence in an online fashion. In this paper, we propose a highly robust solution to address this problem.

A. Contributions

Our solution is motivated by the emerging theory of Robust PCA (RPCA) [3], [28]. In particular, RPCA provides a unified solution to estimating low-rank matrices in the cases of both missing data *and* random data corruption [3]. The algorithm is guaranteed to converge to the global optimum if the ambient space dimension is sufficiently high. Compared to other existing solutions such as incremental SVD, RANSAC, and BPDN, the set of heuristic parameters one needs to tune is also minimal. Furthermore, recent progress in convex optimization has led to very efficient numerical implementation of RPCA with the computational complexity comparable to that of classical SVD [13].

Our proposed solution to online 3-D motion registration consists of two steps. In the initialization step, RPCA is used to estimate a low-rank representation of the rigid-body motion within the first several image frames, which establishes a global shape model of the rigid body. In the online update step, we propose a sparse subspace projection method that projects new observations onto the low-dimensional shape model, simultaneously correcting possible sparse data corruption. The overall algorithm is called *Sparse Online Low-rank projection and Outlier rejection* (SOLO), which is illustrated in Figure 1.

Compared to the popular method of RANSAC, one major benefit of the new solution is that by enforcing a low-rank shape model, those sparsely corrupted image features can be compensated instead of simply being discarded. Although in this paper, we apply the algorithm to 3-D motion features (including general 3-D rigid bodies and degenerate 2-D planar structures) that are tracked by the Microsoft Kinect motion sensor, the same algorithm could also help address the more traditional RBMR problems with 2-D image features. Through extensive simulation and a real-world experiment, we demonstrate that SOLO solves the online RBMR problem with state-of-the-art accuracy and more importantly with improved speed of one to two orders of magnitude faster than RANSAC.

II. 3-D Feature Tracking

In this section, we briefly describe the 3-D feature tracking methodology used in this paper. In our 3-D tracking subsystem (e.g., on Microsoft Kinect), we first identify salient image features, and then track them frame by frame in image space (as an example shown in Figure 2). The features are then reprojected onto the camera coordinate system using depth measurements. Over time, new features are extracted on periodic intervals to maintain a dense set over the image geometry. Each feature is tracked independently, and may be dropped once it leaves the field of view or produces spurious results (jumps) in camera space.

Fig. 2. Tracking results of an indoor scene shown on the first frame of the sequence.

For tracking, we use the Kanade-Lucas-Tomasi feature tracker (KLT) [18]. It is well known that the KLT tracker is extremely fast and can run in real time on a standard desktop computer. For KLT to work effectively, the extracted features must exhibit local saliency. To achieve this and produce a dense set of features over scenes, we use the Harris corner detector as well as a Difference of Gaussians (DoG) extractor [23]. Only the lowest two levels of the DoG pyramid are used. This ensures that the features exhibit high local saliency in a small window and are spatially well-localized.

One implicit advantage of tracking features across multiple frames is that it permits the tracking data to be represented naturally as a matrix. Each (sample-indexed) row represents observations of multiple features in a single time step, while each column represents the observations of a feature over time. Overall, the tracking system we employ demonstrates that simple, efficient algorithms can track well-localized feature trajectories over multiple frames. Together with the registration algorithm described in Section III, our complete system could be deployed on low-cost embedded devices.

As a comparison, most existing feature tracking and motion registration algorithms employ robust feature matching on a frame-by-frame basis [9]. This technique works quite well because RANSAC rejects misaligned features. However, as we will demonstrate in this paper, matching features only between frames neglects the continuity of spatial constraints of these features, which leads to inferior performance. In addition, due to the complexity of these random sampling algorithms, their real-time performance is typically achieved by either using dedicated hardware acceleration [19] or limiting in small-scale SLAM scenarios with sparse feature sets [6] or simplified ground geometric models [12].

III. Online 3-D Rigid Body Motion Registration

A. Problem Statement

First, we shall formulate the 3-D RBMR problem and introduce the notation we will use for the rest of the paper. We denote $\boldsymbol{x}_{i,j} \in \mathbb{R}^3$ as the coordinates of feature j in the ith frame, where $i \in [1, \cdots, F]$ and $j \in [1, \cdots, m]$. In the noise-free case, when the same jth feature is observed in two different frames 1 and i, its images satisfy a rigid-body constraint:

$$\boldsymbol{x}_{i,j} = R_i \boldsymbol{x}_{1,j} + T_i \in \mathbb{R}^3, \tag{1}$$

where $R_i \in \mathbb{R}^{3\times 3}$ is a rotation matrix and $T_i \in \mathbb{R}^{3\times 1}$ is a 3-D translation. This relation can be also written in homogeneous coordinates as

$$\boldsymbol{x}_{i,j} = \Pi \begin{bmatrix} R_i & T_i \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{1,j} \\ 1 \end{bmatrix} \doteq \Pi g_i \begin{bmatrix} \boldsymbol{x}_{1,j} \\ 1 \end{bmatrix}, \tag{2}$$

where $\Pi = [I_3, \mathbf{0}] \in \mathbb{R}^{3\times 4}$ is a projection matrix.

In the noise-free case, since all the features in the ith frame satisfy the same rigid-body motion, one can stack the image coordinates of the same feature in the F frames in a long vector form, and then the collection of all the m features form a data matrix X, which can be written as the product of two rank-4 matrices:

$$X \doteq \begin{bmatrix} \boldsymbol{x}_{1,1} & \cdots & \boldsymbol{x}_{1,m} \\ \vdots & \cdots & \vdots \\ \boldsymbol{x}_{F,1} & \cdots & \boldsymbol{x}_{F,m} \end{bmatrix} = \begin{bmatrix} \Pi g_1 \\ \vdots \\ \Pi g_F \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{1,1}, & \cdots, & \boldsymbol{x}_{1,m} \\ 1, & \cdots, & 1 \end{bmatrix} \in \mathbb{R}^{3F\times m}. \tag{3}$$

In particular, $g_1 = I_4$ represents the identity matrix. It was observed in [21], [14] that when $F, m \gg 4$, the rank of matrix X that represents a rigid-body motion in space is at most four, which is upper bounded by the rank of its two factor matrices in (3). In SfM, the first matrix on the right hand side of (3) is called a *motion matrix* M, while the second matrix is called a *shape matrix* S. Although (3) is not a unique rank-4 factorization of X, a canonical representation can be determined by imposing additional constraints on the shape of the object [21], [14].

Lastly, for motion registration, if we denote the 3-D coordinates (e.g., under the world coordinates centered at the camera) of the first frame as: $W_1 \doteq [\boldsymbol{x}_{1,1}, \cdots, \boldsymbol{x}_{1,m}] \in \mathbb{R}^{3\times m}$, then the rigid-body motion (R_i, T_i) of the features from the world coordinates to any ith frame satisfies the following constraint:

$$W_i \doteq [\boldsymbol{x}_{i,1}, \cdots, \boldsymbol{x}_{i,m}] = R_i W_1 + T_i \mathbf{1}^T. \tag{4}$$

Using (4), the two transformations R_i and T_i can be recovered by the Orthogonal Procrustes (OP) method [17]. More specifically, let $\mu_i \in \mathbb{R}^3$ be the mean vector of W_i, and denote $\bar{W}_i$ as the centered feature coordinates after the mean is subtracted. Suppose the SVD of $\bar{W}_i \bar{W}_1^T$ gives rise to:

$$(U, \Sigma, V) = \text{svd}(\bar{W}_i \bar{W}_1^T). \tag{5}$$

Then the rotation matrix $R_i = UV^T$, and the translation $T_i = \mu_i - R_i \mu_1$.

In this work, we consider a fast online solution to RBMR. Our goal is to maintain the estimation of a low-rank representation of X and its subsequent new observations W_i with minimal computational complexity. Clearly, the accuracy of the OP algorithm depends on the estimation of W_i, especially when the feature data are often affected by moderate amounts of measurement noise, outliers, and missing observations. In the rest of the section, we first discuss the initialization step to jump start the robust low-rank estimation of the initial observations X using Robust PCA in Section III-B. Then we propose a new solution to update the low-rank estimation in the presence of new observations in ith frame W_i in Section III-C. Finally, applying our algorithm on real-world data may encounter additional nuisances such as new feature tracks entering the scene and missing data. After the summary of Algorithm 1, we will briefly show that the proposed solution can be easily extended to handle these additional conditions in an elegant way. Arguably our solution is more robust than the existing frame-by-frame robust techniques in motion registration.

B. Initialization via Robust PCA

In the initialization step, a robust low-rank representation of X needs to be obtained in the presence of moderate Gaussian noise, data corruption, and outlying image features. The problem can be solved *in closed form* by Robust PCA [3], [28]. Here we model $X \in \mathbb{R}^{n\times m}$ as the sum of three components:

$$X = L_0 + D_0 + E_0, \tag{6}$$

where L_0 is a rank-4 matrix that models the ground-truth distribution of the inlying rigid-body motion, D_0 is a Gaussian noise matrix that models the dense noise independently distributed on the X entries, and E_0 is a sparse error matrix that collects those nonzero coefficients at a sparse support set of corrupted data, outlying image features and bad tracks.

The matrix decomposition in (6) can be successfully solved by a *principal component pursuit* (PCP) program:

$$\min_{L,E} \|L\|_* + \lambda \|E\|_1 \quad \text{subj. to} \quad \|X - L - E\|_F \leq \delta, \tag{7}$$

where $\|\cdot\|_*$ denotes matrix nuclear norm, $\|\cdot\|_1$ denotes entry-wise ℓ_1-norm for both matrices and vectors, and λ is a regularization parameter that can be fixed as $\sqrt{\max(n, m)}$. It has been shown in [3], [28] that when the dimension of

matrix X is sufficiently high and with some extra mild conditions on the coefficients of L_0 and E_0, with overwhelming probability, the global (approximate) solution of L_0 and E_0 can be recovered.

The key characteristics of the PCP algorithm are highlighted as follows: Firstly, the regularization parameter λ does not necessarily rely on the level of corruption in E_0, so long as their occurrences are bounded. Secondly, although the theory assumes the sparse error should be randomly distributed in X, the algorithm itself is surprisingly robust to both sparse random corruption and highly correlated outlying features as a small number of column vectors in X. Finally, although the original implementation of PCP in [3] is computationally intractable for real-time applications, its most recent implementation based on an augmented Lagrangian method (ALM) has significantly reduced its complexity [13]. In this paper, we adopt the ALM solver for Robust PCA, whose average run time is merely a small constant (in general smaller than 20) times the run time of SVD. In our online formulation of SOLO, this calculation only needs to be performed once in the initialization step.

Since the resulting low-rank matrix L may still contain entries of outlying features, an extra step needs to be taken to remove those outliers. In particular, one can calculate the ℓ_0-norm of each column in $E_0 = [\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_m]$. With respect to an outlier threshold τ, if $\|\boldsymbol{e}_i\|_0 > \tau$, then $\boldsymbol{e}_i$ represents dense corruption on the corresponding feature track and hence should be regarded as an outlier.[1] Subsequently, the indices of the inliers define a support set $I \subset [1, \cdots, m]$. Hence, we denote the cleaned low-rank data matrix after outlier rejection as

$$\hat{L} \doteq L^{(I)}. \tag{8}$$

Finally, we note that although in (7), L represents the optimal matrix solution with the lowest possible rank, due to additive noise and data corruption in the measurements, its rank may not necessarily be less than five. Therefore, to enforce the rank constraint in the RBMR problem and further obtain a representative of the shape matrices that span the 4-D subspace, an SVD is performed on $\hat{L}$ to identify its right eigenspace:

$$(U, \Sigma, V) = \text{svds}(\hat{L}, 4), \tag{9}$$

where $V^T \in \mathbb{R}^{4\times m}$ is then a representative of the rigid body's shape matrices.

C. Sparse Online Low-rank projection and Outlier rejection (SOLO)

In this section, we propose a novel algorithm that projects new observations W_i from the ith frame onto the rigid-body shape subspace. This subspace is parameterized by the shape matrix V^T that we have estimated in the initialization step.[2] Traditionally, a (least squares) subspace projection operator would project a (noisy) sample perpendicular to the surface of the subspace that it is close to, which only involves basic matrix-vector multiplication. However, in anticipation of continual random feature corruption during the course of feature tracking for RBMR, the projection must also be robust to sparse error corruption in W_i. Hence, we contend that SOLO is a more appropriate yet still efficient algorithm to achieve online motion registration update.

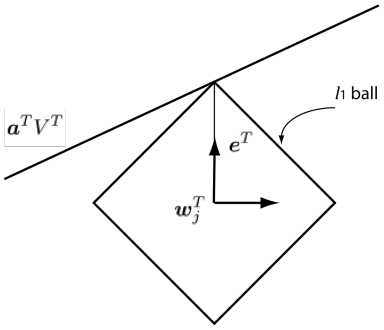


Fig. 3. A visualization of sparse subspace projection as basis-pursuit denoising, which can be solved by ℓ_1-minimization.

Given the initialization $\hat{L}$ and the inlier support set I, without loss of generality, we assume W_i only contains those features in the support set I. [3] As discussed in (3) and (9), matrix V^T from the SVD of $\hat{L}$ is a representative of the class of all the shape matrices of the rigid body up to an ambiguity of 4-D rotation on the subspace. Therefore, the new observations W_i of the same features should also lie on the same shape subspace. That is, let $W_i = [\boldsymbol{w}_1^T; \boldsymbol{w}_2^T; \boldsymbol{w}_3^T]$, where each $\boldsymbol{w}_1^T \in \mathbb{R}^{1\times m}$ is a row vector. Then

$$\boldsymbol{w}_j^T = \boldsymbol{a}^T V^T \quad \text{for some} \quad \boldsymbol{a}^T \in \mathbb{R}^{1\times 4}. \tag{10}$$

In the presence of sparse corruption, the row vector $\boldsymbol{w}_j^T$ is perturbed by a sparse vector $\boldsymbol{e}$:

$$\boldsymbol{w}_j^T = \boldsymbol{a}^T V^T + \boldsymbol{e}^T, \quad \text{where } \boldsymbol{e}^T \in \mathbb{R}^{1\times m}. \tag{11}$$

The sparse projection constraint (11) bears resemblance to BPDN in compressive sensing literature [5], as a sparse error perturbs a high-dimensional sample away from a low-dimensional subspace model. The standard procedure of BPDN using ℓ_1-minimization (ℓ_1-min) is illustrated in Figure 3.

However, we notice that a BPDN-type solution via ℓ_1-min may not be the optimal solution to our problem. The reason is that the row vectors in $W = [\boldsymbol{w}_1^T; \boldsymbol{w}_2^T; \boldsymbol{w}_3^T]$ are not three arbitrary vectors in the 4-D subspace V^T. In fact, the three vectors must be projected onto a nonlinear manifold M embedded in the shape subspace V^T, and the span of the shape model can be interpreted as the linear hull of the feasible rigid-body motions between W_1 and W_i. Figure 4 illustrates this rigid-body constraint applied to sparse subspace projection in 3-D.

[1] For those coefficients in $\boldsymbol{e}_i$ with small nonzero values, a hard-thresholding can be applied to reduce the values to zero.

[2] In this paper, we may choose to abuse the notation of V^T to also represent the 4-D subspace.

[3] In this paper, we do not address the SLAM scenario where the sequence could include new 3-D structures. In that case, one can simply re-initialize an updated shape matrix V as in the previous section.

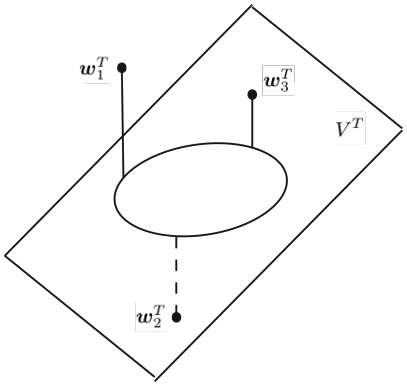


Fig. 4. The row vectors of W should be projected onto a manifold in V^T that represents a valid rigid-body motion.

Our algorithm of *sparse shape subspace projection* is described as follows. Given the observation W_i and a shape subspace V^T, the algorithm minimizes:

$$\min_{E,A} \|E\|_1 \quad \text{subj. to} \quad W_i = AV^T + E. \tag{12}$$

By virtue of low dimensionality of this hull, together with the sparsity of the residual, the projected data AV^T should be well localized on the manifold. Hence, in addition to being consistent with a realistic (sparse) noise model, the new sparse subspace projection algorithm (12) also implies the benefit of good localization in the motion space.

The objective can be solved quite efficiently (and much faster than solving RPCA in the initialization) by the same augmented Lagrangian approach in [13]:

$$\min_{A,E,Y,\mu} \|E\|_1 + \langle Y, W_i - AV^T - E\rangle + \frac{\mu}{2}\|W_i - AV^T - E\|_F^2, \tag{13}$$

where Y is a matrix of Lagrange multipliers, and $\mu > 0$ represents a monotonically increasing penalty parameter during the optimization. The optimization only involves a soft-thresholding function applied to the entries of E and matrix-matrix multiplication for the update of A and E, and does not involve computation of singular values as in RPCA.

Finally, the rigid-body motion between each W_i and the first reference frame W_1 after the projection can be recovered by the OP algorithm (5). However, as the projection (12) may be also affected by dense Gaussian noise, the estimated low-rank component may not accurately represent a consistent rigid-body motion. As a result, what we can do is to identify an index set I_i for those uncorrupted features with zero coefficients in E. The OP algorithm will be applied only using the uncorrupted *original* features in W_1 and W_i. In a sense, this motion registration algorithm resembles the strategy in RANSAC to select inlying sample sets. However, our algorithm has the ability to directly identify the corrupted features via sparse subspace projection, and hence the process is noniterative and more efficient.

It is also improtant to emphasize that the algorithm overcomes sensitivity to error in the calculation of W_1 by several mechanisms. First, W_1 is obtained by RPCA, ensuring a robust estimate even when a sample of the shape is corrupted in the *first observation*. Second, samples of W_1 with fully corrupted tracks are identified in the sparse error term and ignored during registration. Most importantly, the estimate of W_1 can be refined efficiently based on incoming data using RPCA on the concatenated data matrix.

The complete algorithm, *Sparse Online Low-rank projection and Outlier rejection* (SOLO), is summarized in Algorithm 1.

Algorithm 1: SOLO

Input: Initial observations X, feature coordinates of the reference frame W_1, and W_i for each subsequent frame i.

1: **Init:** Compute L and I of X via RPCA (7).
2: $W_1 \leftarrow W_1^{(I)}$, remove outliers in the reference frame.
3: $[U, \Sigma, V] = \text{svds}(L^{(I)}, 4)$.
4: **for** Each new observation frame i **do**
5: $W_i \leftarrow W_i^{(I)}$.
6: Identify corruption E via sparse subspace projection (12).
7: Let I_i be the index set of uncorrupted features in W_i.
8: Estimate (R_i, T_i) using inlying samples in $I_1 \cap I_i$.
9: **end for**

Output: Inlier support set I, rigid-body motions (R_i, T_i).

Before we proceed to discuss results from our experiment, it is worth mentioning a straightforward yet elegant extension of the algorithm in the presence of missing data. In the initialization step, one can rely on a variant of RPCA to recover the missing data in matrix X. The technique is known as *low-rank matrix completion* [2], [3], which minimizes a similar low-rank representation objective constrained on the observable coefficients:

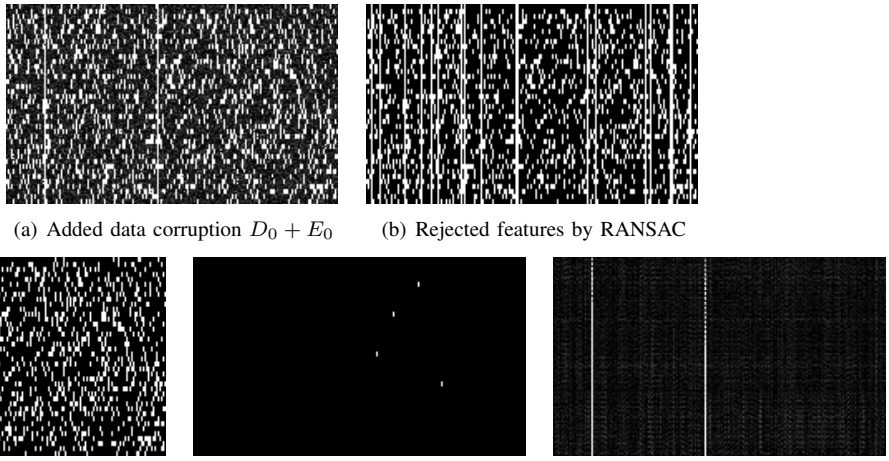
$$\min_{L,E} \|L\|_* + \lambda\|E\|_1 \quad \text{subj. to} \quad \mathcal{P}_\Omega(L+E) = \mathcal{P}_\Omega(X), \tag{14}$$

where Ω is an index set of those features that remain visible in X, and $\mathcal{P}_\Omega$ is the orthogonal projection onto the linear space of matrices supported on Ω.

Using low-rank matrix completion (14), in the presence of a partial measurement of new feature tracks, those incomplete new observations should be identified as tracks with missing data. Then a new initialization step using (14) should be performed on a new data matrix X that includes the new tracks to re-establish the shape subspace and inlier support set I as in (9).

IV. Experiment

In this section, we validate the performance of SOLO algorithm and compare with the classical RANSAC solutions, which has been the most popular solution to date for SLAM and motion registration. In the rest of the section, the two algorithms will be applied to a thorough list of simulations and a real-world experiment. The benchmarks are calculated



(a) Added data corruption $D_0 + E_0$ (b) Rejected features by RANSAC

(c) Estimated sparse error by RPCA (d) Sparse error difference $|E_0 - E^*|$ (e) Ground-truth difference $|L_0 - L^*|$

Fig. 5. A visualization of the estimation error of a simulated motion matrix X by RANSAC and RPCA. The added data corruption and estimation difference are represented as white pixels in the images.

on a 2 GHz PC with an Intel Core i7 processor and in MATLAB environment.

A. Simulated Analysis

We first use synthesized data to benchmark the accuracy and speed of our batch motion registration algorithm described in Section III-B. The calculation of (R_i, T_i) between each pair of W_1 and W_i will be based on $\hat{L}$ alone as the output of RPCA and outlier rejection (8). We compare the performance of motion registration by RPCA with that by the classical solution of RANSAC on a frame-by-frame basis. The minimal feature set in RANSAC is set to four.

In one simulation, the outlier rejection results in motion registration by RPCA and RANSAC are visualized in Figure 5. In this example, we observe that RPCA is much more effective in identifying both random data corruption and outlying feature tracks (that post inconsistent feature measurements in the entire columns) than RANSAC. Also note that the large coefficient difference in the two columns of Figure 5(e) should not be a concern, as it is well known that RPCA cannot uniquely recover dense column corruption [3], and nevertheless the corresponding features will be rejected as outliers by (8). Finally, we can also see quite significant difference between the ground-truth low-rank matrix L_0 and its estimate L^*. It shows the accuracy of RPCA is still sensitive to high variance dense Gaussian noise.

To overcome the issue of dense Gaussian noise in RPCA, our recommended implementation further adds a RANSAC-style refinement stage, which selects a minimal set of inlying samples from the support set already identified by RPCA. Correspondences consistent with the constructed model are merged until the refinement stage converges. Typically this recursive refinement process converges in 2–4 iterations. With this in mind, we show the accuracy and speed of motion registration using RPCA and RANSAC in Figure 6. In Figure 6(d) the motion registration accuracy w.r.t. two matrices (R, T) is measured by the sum of the difference to the ground truth (R_0, T_0) in Frobenius norm.

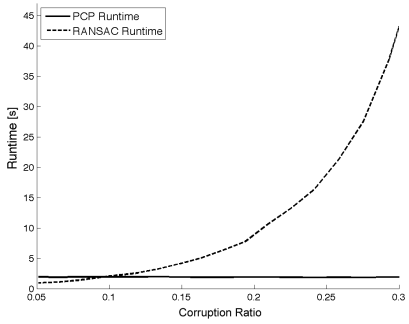
We can see in Figure 6(a), with certain level of accuracy confidence, the average runtime of RANSAC grows super-linearly with the increase of the corruption percentage, while RPCA remains effective in compensating those corruptions in the low-rank matrix. Figure 6(b) and 6(c) show reasonable increase in computation time for RPCA w.r.t. the number of features and the number of frames in the motion window X. Finally, the accuracy about the estimated rigid-body transformation is shown in Figure 6(d). Without the additional refinement stage, RPCA already achieves comparable result than RANSAC. If the iterative refinement is added to the algorithm, we can see significant improvement in the estimation of the motion. Notice that the estimation errors of R and T are already very small in all three cases, as shown on the y-axis.

B. Performance on Kinect Data

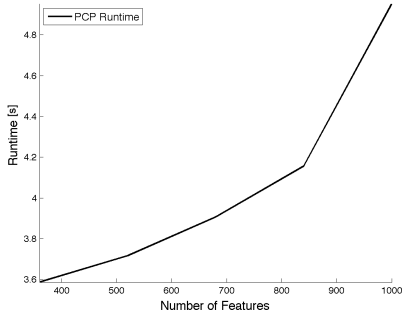
We now test the performance of the online SOLO algorithm combined with the KLT tracker on a set of real-world depth data collected by a Microsoft Kinect sensor. The data are collected in an indoor lab environment. In total, there are 50 frames tracked and the first 25 frames are used for initialization. The motion registration and scene reconstruction results are shown in Figure 7.

In our experiment, we found the KLT tracking scheme applied on Kinect to be highly effective in practice, producing upwards of two hundred tracks in a typical indoor setting. This ensures that the initialization of X has enough features to converge to the correct low-rank model L^*. As expected, the KLT feature tracker also produces small amounts of local jumps due to repetitive object textures (e.g., the checkerboard pattern).

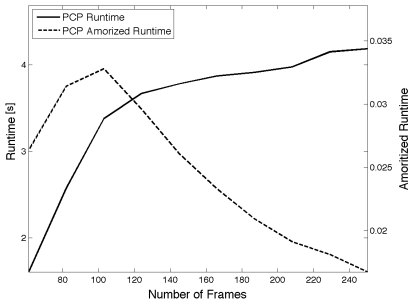
For this experiment, we have tuned RANSAC specifically for the empirical sample corruption ratio in the scene. Despite this effort, SOLO is still faster by a factor of two compared to RANSAC. We emphasize that oracle tuning provides a lower bound on the complexity of RANSAC, and



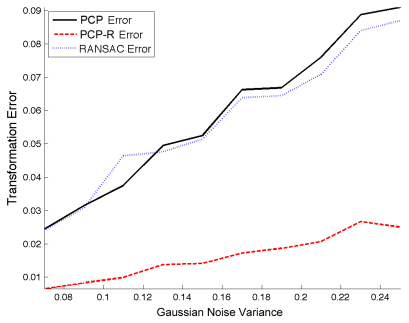
(a) Average runtime vs. corruption percentage



(b) RPCA runtime

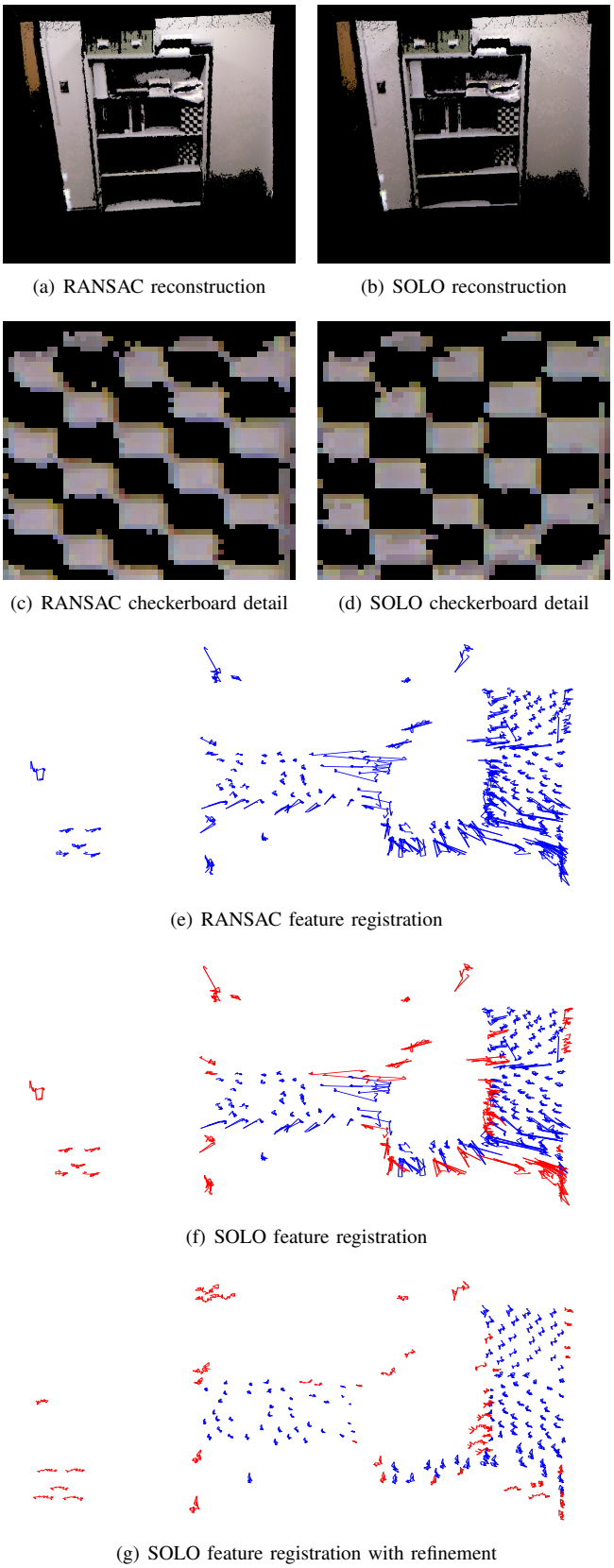


(c) RPCA runtime vs. number of frames (with 250 features)



(d) Motion registration accuracy vs. noise variance

Fig. 6. A simulated comparison between RPCA and RANSAC. PCP is based on the ALM method. PCP-R means the RPCA approach with a RANSAC-style iterative refinement stage.



(a) RANSAC reconstruction (b) SOLO reconstruction

(c) RANSAC checkerboard detail (d) SOLO checkerboard detail

(e) RANSAC feature registration

(f) SOLO feature registration

(g) SOLO feature registration with refinement

Fig. 7. A comparison of SOLO- and RANSAC-based registration results for a real world 3-D reconstruction problem. (e)-(g) are feature trajectories in the world coordinate system. Features discarded by our algorithm are shown in red.

its complexity would be much higher in a less-controlled, online setting.

The enlarged checkerboard references demonstrate crisper results for the SOLO registration than the RANSAC reg-

istration. More interestingly, Figures 7(e)–7(g) demonstrate feature registrations for RANSAC and SOLO. The red trajectories are those which are selected by SOLO for rejection. Despite spurious recorded behavior, such as coarse spatial discontinuities, many of the tracks are salvageable and properly localized in the cleaned data L^*. Overall, SOLO demonstrates equally good or better registration quality than RANSAC, if measured qualitatively.

V. Conclusion and Discussion

We have proposed an online 3-D motion registration algorithm called SOLO. Its main advantage compared to existing robust statistical methods such as RANSAC is that the algorithm is capable of exploiting the underlying low-rank matrix structure in describing the motion and shape of a dominant rigid body. The initialization step employs Robust PCA to recover such low-rank matrices and compensate gross feature corruption and outliers. The online update step sequentially projects new observations onto the inlier shape subspace by a sparse subspace projection technique, which is efficient to implement as a convex program. In our extensive experiment, we have demonstrated equally good or better motion registration accuracy compared to RANSAC, with significant speed-up by one to two orders of magnitude.

For future problems, the convincing results shown in the paper can bring SOLO to a broader range of applications in SLAM. In this paper, we have considered the motion registration problem for a single motion. In a more complex dynamic scene, multiple motions may be captured by the 3-D camera. In addition, the multiple motions may be either independent or constrained (e.g., a humanoid robot consists of multiple linked rigid limbs and the torso). These are some of the interesting problems we intend to investigate further. We believe the SOLO framework has laid a solid foundation for us to tackle these problems.

References

[1] H. Aanaes, R. Fisker, K. Astrom, and J. Carstensen. Robust factorization. *PAMI*, 24(9):1215–1225, 2002.

[2] J. Cai, E. Candes, and Z. Shen. A singular value thresholding algorithm for matrix completion. *arXiv:0810.3286*, 2008.

[3] E. Candes, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journals of the ACM*, 58(1):1-37, 2009.

[4] P. Chen and D. Suter. Recovering the missing components in a large noisy low-rank matrix: Application to SFM. *PAMI*, 26(8):1051–1063, 2004.

[5] S. Chen, D. Donoho, and M. Saunders. Atomic decomposition by basis pursuit. *SIAM J. Sci. Comput.*, 20(1):33–61, 1998.

[6] A. Davison, I. Reid, N. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *PAMI*, 29(6):1052–1067, 2007.

[7] M. Fischler and R. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[8] R. Hartley. PowerFactorization: an approach to affine reconstruction with missing and uncertain data. In *Australia-Japan adv. workshop on computer vision*, 2003.

[9] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox. RGB-D mapping: using depth cameras for dense 3D modeling of indoor environments. In *Int. Symp. Experimental Robotics*, 2010.

[10] F. Kahl and A. Heyden. Affine structure and motion from points, lines and conics. *IJCV*, 33(3):163–180, 1999.

[11] K. Kanatani. Motion segmentation by subspace separation and model selection. In *ICCV*, 2001.

[12] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *International Smposium on Mixed and Augmented Reality*, 2007.

[13] Z. Lin, M. Chen, L. Wu, and Y. Ma. The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices. *Dept. of Elec. and Comp. Engr., Univ. Illinois-UC, Tech. Rep*, No. UILU- ENG-09-2215, 2009.

[14] C. Poelman and T. Kanade. A paraperspective factorization method for shape and motion recovery. *PAMI*, 19(3):206–218, 1997.

[15] Q. Ke and T. Kanade. Robust ℓ_1 norm factorization in the presence of outliers and missing data by alternative convex programming. In *CVPR*, 2005.

[16] K. Saeki, K. Tanaka, and T. Ueda. LSH-RANSAC: An incremental scheme for scalable localization. In *ICRA*, 2009.

[17] P. Schonemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, 1966.

[18] J. Shi and C. Tomasi. Good Features to Track, In *CVPR*, 1994.

[19] S. Sinha, J. Frahm, M. Pollefeys, and Y. Genc. Feature tracking and matching in video using programmable graphics hardware. *Machine Vision and Applications*, 22:207–217, 2011.

[20] C. Stewart. Robust parameter estimation in computer vision. *SIAM Review*, 41(3):513–537, 1999.

[21] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *IJCV*, 9(2):137–154, 1992.

[22] P. Torr and C. Davidson. IMPSAC:synthesis of importance sampling and random sample consensus. *PAMI*, 25(3):354–364, 2003.

[23] T. Tuytelaars and K. Mikolajczyk. Local invariant feature detectors: a survey. *Foundations and Trends in Computer Graphics and Vision*, 3(3):177–277, 2008.

[24] R. Vidal and Y. Ma. A unified algebraic approach to 2-D and 3-D motion segmentation and estimation, *J. Math. Imaging & Vision*, 25:403–421, 2006

[25] H. Wang and D. Suter. MDPE: a very robust estimator for model fitting and range image segmentation. *IJCV*, 59(2):139–166, 2004.

[26] B. Williams, P. Smith, and I. Reid. Automatic relocalisation for a single-camera simultaneous localisation and mapping system. In *ICRA*, 2007.

[27] A. Yang, S. Rao, and Y. Ma. Robust statistical estimation and segmentation of multiple subspaces. In *CVPR Workshop on 25 Years of RANSAC*, 2006.

[28] Z. Zhou, J. Wright, X. Li, E. Candes, and Y. Ma. Stable principal component pursuit. In *ISIT*, 2010.