

Simultaneous Compaction and Factorization of Sparse Image Motion Matrices

Susanna Ricco and Carlo Tomasi

Department of Computer Science
Duke University
Durham, NC USA
`{sricco,tomasi}@cs.duke.edu`

Abstract. Matrices that collect the image coordinates of point features tracked through video – one column per feature – have often low rank, either exactly or approximately. This observation has led to many matrix factorization methods for 3D reconstruction, motion segmentation, or regularization of feature trajectories. However, temporary occlusions, image noise, and variations in lighting, pose, or object geometry often confound trackers. A feature that reappears after a temporary tracking failure – whatever the cause – is regarded as a new feature by typical tracking systems, resulting in very sparse matrices with many columns and rendering factorization problematic. We propose a method to simultaneously factor and compact such a matrix by merging groups of columns that correspond to the same feature into single columns. This combination of compaction and factorization makes trackers more resilient to changes in appearance and short occlusions. Preliminary experiments show that imputation of missing matrix entries – and therefore matrix factorization – becomes significantly more reliable as a result. Clean column merging also required us to develop a history-sensitive feature reinitialization method we call *feature snapping* that aligns merged feature trajectory segments precisely to each other.

1 Introduction

Many problems in computer vision require tracking point features through video sequences. Classical examples include three-dimensional geometric reconstruction, motion segmentation, and motion compression. Features are distinguished from each other by the appearance of small image windows centered around the points of interest, and are tracked by correlating descriptors of appearance across consecutive frames. The resulting image coordinates are often arranged in a measurement matrix M with one column per point. If the world is simple – perhaps characterized by one or a few rigid motions, or by an articulated body – the matrix M has been shown to be low rank, and its factorization $\arg \min_{L,R} \|M - LR^T\|$ in some norm yields important information about the scene and its motions.

However, tracking point features is brittle, because the appearance within a feature window can change as a result of many factors including image noise, variations in lighting or viewpoint, or object deformations. In addition, the window being tracked may become occluded (hidden from view), perhaps to reappear several frames later. Although many tracking failures are short-lived, a reappearing feature is typically viewed as an entirely new feature, and a new column is added for it to the matrix M . This causes M to be extremely sparse, with only a small fraction of known entries in each column, and makes finding a good factorization a challenge.

In this paper, we propose a method to simultaneously factor and compact the measurement matrix M into a smaller and denser matrix \hat{M} by merging groups of columns of M that correspond to the same feature into single columns of \hat{M} . Specifically, we formulate a mixed integer program that factors M while simultaneously merging together columns that are temporally, photometrically and geometrically consistent with each other. Two columns are temporally consistent with each other if their known entries come from disjoint time intervals. They are photometrically consistent if the corresponding feature windows look similar to each other. The two columns are geometrically consistent if the trajectories that they represent align well in the image. The resulting integer program is intractable, and we solve it approximately by alternating between factorization and compaction until convergence. In so doing, we lose guarantees of convergence to a global optimum, but our experiments show satisfactory results nonetheless.

Two (or more) columns can be merged when they correspond *exactly* to the same point. However, standard trackers initialize new features independently of past history, so it is unlikely that a new feature is found to coincide perfectly with an old one. To address this difficulty, we propose a history-sensitive feature reinitialization method, in which the image coordinates of newly defined features are *snapped* whenever possible to image positions that maximize the photometric and geometric similarity with a previously seen feature.

The proposed method yields a compacted matrix \hat{M} that is denser than its original counterpart M . Our preliminary experiments show that this greater density in turn leads to better generalization, in the sense that the missing entries that the compact factorization $L\hat{R}^T$ imputes in \hat{M} are more accurate than those that the sparse factorization LR^T imputes in M .

In summary, our main contributions are (1) a method for coupling matrix compaction and factorization using chronological, geometric and photometric evidence, and (2) a history-sensitive feature reinitialization method we call *feature snapping* that precisely aligns merged feature trajectories to each other.

The rest of the paper is organized as follows. After a review of related literature, Section 3 presents our formulation of the combined factorization and matrix compaction problem. Section 4 discusses our technique for solving the resulting optimization problem. Section 5 describes feature snapping, and the experimental results in Section 6 show the benefits of our method. Section 7 concludes with summary, a discussion of limitations, and future work.

2 Related Work

Matrix factorization with missing data is an active area of research both within and outside computer vision. For example, collaborative filtering problems such as the Netflix Prize [1] require imputing missing entries of a measurement matrix and assume that the latter is low rank. In this paper, we focus on prior work on matrix factorization for tracking or Structure-from-Motion (SfM).

Tomasi and Kanade [2] introduced the original factorization formulation for SfM of a rigid object. Since then, it has been extended to multi-body [3] and non-rigid [4, 5] motions. Early approaches often ignored the problem of missing data, or greedily stitched together a solution to the full problem by considering a number of smaller problems on full sub-blocks of the measurement matrix.

Later research focused on the full missing-data factorization problem $M \approx LR^T$ without using imputed values for missing entries during optimization. Techniques can be separated into *alternation* [5–7], where one iterates between solving for L with R fixed and vice versa, and *non-linear minimization* that considers all unknowns simultaneously. Buchanan and Fitzgibbon [8] provide an excellent summary of many alternation techniques and present a damped Newton method for factorization with missing data.

Many of these techniques find what is believed to be the globally optimal solution on standard datasets (although convergence to the global optimum is not guaranteed), but exhibit poor generalization performance because rank constraints alone are insufficient to constrain the problem fully. This implies that the objective function must be modified. In this vein, Buchanan and Fitzgibbon report experiments where a prior that enforces orthonormality of the motion factor L increases the per-pixel reconstruction error on the observed entries but results in visually more satisfactory tracks in terms of generalization performance. Torresani and Hertzmann [9] embed the problem in a probabilistic framework and propose adding priors on both L and R to improve performance. Gotardo and Martinez [10] and Olsen and Bartoli [11] add temporal smoothness priors by either including an explicit penalty on the temporal derivative of columns of L [11], or by building L from a pre-defined basis with a desired level of smoothness [10]. Del Bue *et al.* [12] and Olsen and Bartoli [11] also include shape priors on the matrix R .

Surprisingly, the SfM literature rarely considers the opportunity for *matrix compaction*, in which different columns that correspond to the same feature are merged together during factorization. In most prior work, it is assumed that correct correspondences are provided by the tracker throughout. Some work on robust matrix factorization [11, 13] allows for individual correspondences to be flagged as outliers and discarded. This is the reverse of the problem we focus on, as we look for tracks that should have been merged and were not, rather than associations that were made and should not have. Work by Dellært *et al.* [14] is a notable exception. The authors assume that no correspondences are known at all, and propose a Monte Carlo approach for sampling possible correspondences and then solve for structure and motion using the full-perspective bundle adjustment formulation [15]. Our approach differs in three ways. First, we assume that some,

but not all, correspondences are known; that is, we assume that a point-feature tracker often works. We use a more general low-rank matrix model in place of rigidity, making our approach applicable to multi-body or non-rigid motions. Finally, we allow the photometric appearance of tracks in the video to inform the predicted associations, while Dell art *et al.* use geometric information only.

Our work is also related to the ‘‘track gluing’’ experiment reported by Olsen and Bartoli [11]. Their paper considers merging tracks only after the fact and relies only on geometric information to decide when two tracks should be merged into one. In contrast, we interleave matrix compaction with matrix factorization, and include the appearance of tracks in the decision about compaction.

3 Simultaneous Compaction and Factorization

Let M be a $2m \times n$ real *measurement matrix* with missing entries. Starting all indices at 1, the x coordinate of point i in frame f is entry $(2f - 1, i)$ of M , and the corresponding y coordinate is just below, at $(2f, i)$. The term *track i* denotes the trajectory traced by the known entries in column i . Entry (f, i) of an $m \times n$ binary *fill matrix* F is 1 if and only if a measurement for point i is available at frame f , so M has a measurement wherever the $2m \times n$ matrix $F_2 = F \times [1 \ 1]^T$ has a 1. In this expression, ‘ \times ’ denotes the Kronecker product. For convenience, we fill missing entries of the M with a value of zero.

Without compaction, factorization of M can be written as follows:

$$\arg \min_{L,R} \|F_2 \odot (M - LR^T)\|_F^2 \tag{1}$$

where the subscript F denotes the Frobenius norm and ‘ \odot ’ is the Hadamard (*i.e.*, entry-by-entry) product. The two real matrices L and R have sizes $2m \times r$ and $n \times r$, where r is an upper bound on the rank of the ideal measurement matrix.

To express compaction, we introduce a third unknown, the $n \times n$ binary symmetric *compaction matrix* C with entries $c_{ij} = 1$ if and only if track i and j correspond to the same world point and therefore should be merged into the same column. Because the compaction matrix defines an equivalence relationship between track fragments, its entries are required to obey the transitive property. The compact version \hat{M} of M consists of the unique columns of the matrix MC . We encourage compaction by adding a term to the standard factorization objective function (1) that is proportional to the number of zeros in C .

We add additional constraints and heuristics to ensure that the recovered compaction is plausible. First, two tracks i, j are *temporally consistent* if they do not overlap in time. So, the fill matrix F after compaction cannot have entries greater than 1 and C must satisfy the constraint $\|FC\|_\infty = 1$.

Second, two tracks are *photometrically consistent* if small image windows around them look similar. Specifically, let I_i and I_j be the image frames halfway through track i and track j , with $i > j$. We define the $n \times n$, lower-triangular *photometric discrepancy matrix* Φ to collect the sum of squared differences between windows W centered at the coordinates \mathbf{x}_i and \mathbf{x}_j of features i and j in I_i

and I_j after an affine geometric correction [16] to allow for slight deformations over time:

$$\phi_{ij} = \min_{A \in \mathbb{R}^{2 \times 2}, \mathbf{b} \in \mathbb{R}^{2 \times 1}} \sum_W \|I_i(\mathbf{x} - \mathbf{x}_i) - I_j(A(\mathbf{x} - \mathbf{x}_j) + \mathbf{b})\|_F^2 \quad \text{for } i > j. \quad (2)$$

We then encourage photometric consistency by adding to the penalty function in (1) a term proportional to

$$\phi^T \mathbf{c} \quad \text{where } \phi = \text{vec}(\Phi) \quad \text{and} \quad \mathbf{c} = \text{vec}(C). \quad (3)$$

The expression $\text{vec}(C)$ for a matrix denotes the column vector obtained by listing all the entries of C in an arbitrary but fixed order.

Finally, tracks fragments that are merged must be *geometrically consistent*, meaning that the same reconstructed track can approximate the observations from both track fragments. We can enforce this by adding a constraint that $c_{ij}(r_{ik} - r_{jk}) = 0$ for all i, j, k with $i > j$. In words, either track i and j are kept separate so that $c_{ij} = 0$, or they are merged, in which case the corresponding rows of R from (1) must be identical.

In summary: *Simultaneous compaction and factorization of the $2m \times n$ real measurement matrix M amounts to computing the*

$$\arg \min_{L, R, C} \|F_2 \odot (M - LR^T)\|_F^2 + \lambda_1 \mathbf{1}^T(1 - \mathbf{c}) + \lambda_2 \phi^T \mathbf{c} \quad (4)$$

with $L \in \mathbb{R}^{2m \times r}$, $R \in \mathbb{R}^{n \times r}$, $C \in \{0, 1\}^{n \times n}$, and subject to the constraints

$$C = C^T \quad (5)$$

$$(1 - c_{ij}) + (1 - c_{jk}) \geq (1 - c_{ik}) \quad \forall i, j, k \quad (6)$$

$$\|FC\|_\infty = 1, \quad (7)$$

$$c_{ij}(r_{ik} - r_{jk}) = 0 \quad \forall i, j, k. \quad (8)$$

The coefficients $\lambda_{1,2}$ can be found experimentally by cross validation, and ϕ , \mathbf{c} are defined in equation (3).

In this paper, the approximate rank r of M is assumed to be given. If r is not known, our method could be coupled with a model selection process that estimates it. Our experiments with synthetic data suggest that adding compaction during factorization improves performance even when the true rank is overestimated, reducing the importance of the model selection step.

The constraints above ensure that the associations proposed are valid, as discussed earlier. Figure 1 shows an example of a valid compaction for a particular measurement matrix fill pattern.

4 Optimization

The mixed integer quadratically constrained quadratic program (4) is intractable and solving it exactly is impractical for typical problem sizes. We find an approximate solution by alternating between matrix factorization with a fixed compaction and finding a new candidate compaction matrix.

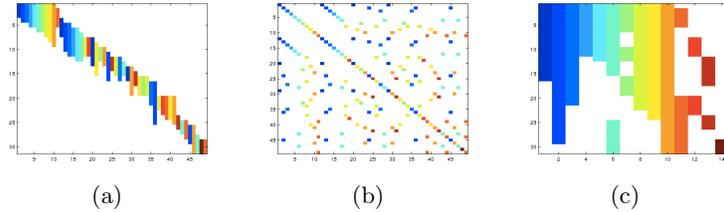


Fig. 1. A typical measurement matrix fill pattern (left). These tracks belong to points on the moving car from the *car-segmented* sequence. White indicates no observation; other colors indicate an observation. Colors in the fill pattern correspond to connected components (columns) of the compaction matrix C (center). Color in this matrix indicates $c_{ij} = 1$, with each connected component artificially colored. After collapsing columns by selecting unique columns of MC (right), we achieve an impressive reduction in the fraction of missing data. The percentage of missing data in the original matrix M is 84% while in the collapsed matrix it is just 45%.

With compaction fixed, only the first term of the objective function and the last constraint remain. We enforce the constraint by collapsing M into \hat{M} and performing factorization on the smaller matrix using a standard method for SfM with missing data. The solution to this step is not unique, as $L\hat{R}^T$ leads to the same penalty value as $LHH^{-1}\hat{R}^T$ where H is any invertible $r \times r$ matrix. This is a standard issue in SfM, and we rely on standard techniques to select a solution. We recover R by duplicating rows of \hat{R} according to the associations defined in the compaction matrix.

With factorization fixed, we replace the geometric consistency constraint and the first term of the objective function with an upper bound on the additional fitting error required if we were to enforce the geometric consistency constraint during the factorization step. We define a lower-triangular *geometric discrepancy matrix* G to have entries

$$g_{ij} = \min(D(i, j), D(j, i)) \quad \text{for } i > j \quad (9)$$

where

$$D^2(i, j) = \sum_f F_2(f, j) (L(f, :)R(i, :)^T - M(f, j))^2 \quad (10)$$

using the Matlab notation $L(f, :)$ to denote row f of matrix L . In words, the distance term $D(i, j)$ is the total distance between the imputed version of track i and the observed version of track j over the frames in which track j is visible. The minimum of $D(i, j)$ and $D(j, i)$ yields a symmetric distance function and represents an additional penalty that would be paid if the factorization procedure selected the best of the two current reconstructions to use for both track fragments. This upper bounds the actual additional penalty that will be paid in the optimal data error with the equality constraints enforced. We solve the integer linear program

$$\arg \min_C \mathbf{g}^T \mathbf{c} + \lambda_1 \mathbf{1}^T (1 - \mathbf{c}) + \lambda_2 \phi^T \mathbf{c} \quad (11)$$

with $C \in \{0, 1\}^{n \times n}$, and subject to the constraints

$$C = C^T \tag{12}$$

$$(1 - c_{ij}) + (1 - c_{jk}) \geq (1 - c_{ik}) \quad \forall i, j, k \tag{13}$$

$$\|FC\|_\infty = 1. \tag{14}$$

We iterate until the proposed compaction matrix C stabilizes, and the solution with the smallest value of the original penalty function from (4) encountered so far is returned. Because of the approximation that replaces the geometric consistency constraint, some compactations may be missed. We search for additional compactations that may have large estimated geometric consistency by greedily performing additional compactations until none are left that decrease the value of the penalty function. Although this procedure does not guarantee a global minimum, our experiments show good results in practice.

5 Feature Snapping

When a standard tracker such as Lucas and Kanade [17] loses a significant fraction of feature points, it is standard practice to start a new set of tracks (thereby adding new columns to M) based on a stateless analysis of the current frame. Points surrounded by sufficient texture are found, and a predefined distance threshold is used to retain points that are far enough from each other and from the live points that are still being tracked [16]. Since this approach to reinitialization only looks at the current frame, new features may be detected at image locations that are *close* to those of features that had been lost earlier and have now reappeared, but not necessarily identical to them. In other words, different tracks to be merged may be slightly misaligned.

To address this difficulty, we introduce a history-sensitive form of feature reinitialization that we call *feature snapping* that positions new tracks preferentially at points that had been previously seen but were lost for a short amount of time. To this end, we maintain a record of the temporally averaged image patch around each live feature. With the record, we also store the current position of the feature. When a track dies, we move its patch and position record to a catalog of lost patches.

At reinitialization time, we first form a list of feature candidates with the traditional, stateless method. We then compute the affine motion [16] between each candidate in the list and the features in the catalog whose stored positions are close to that of the candidate and that were lost for a short amount of time. If such a motion is found with a small photometric residual, the candidate is replaced with the feature from the catalog, that is, the position of the candidate feature is *snapped* to one of the predicted feature positions. Candidate features that cannot be snapped are also retained for further tracking.

Importantly, when snapping occurs, we do *not* assume that a new feature and the one it is snapped to are to be merged. Instead, we leave that decision to the compaction algorithm, which uses the stronger constraints from rank, geometry, and photometric appearance to determine if merging is appropriate.

In future work, we plan to extend the effectiveness of feature snapping to longer occlusion intervals by replacing the stored position of each feature in the catalog with the parameters of a dynamic system estimated from the visible portion of the image trajectory. Combined with a model of growing position uncertainty in the absence of data, this dynamic system allows predicting feature position over longer periods of time.

6 Experiments

We present experimental results on four sequences examining our ability to recover a correct compact matrix \hat{M} and the resulting improved generalization performance. The results presented below were computed using the Column Space Fitting (CSF) algorithm of Gotardo and Martinez [10] as the internal factorization routine and parameters $\lambda_1 = 30$, $\lambda_2 = 1$. Our choice of CSF for factorization is not crucial. We could easily replace it with other standard matrix factorization algorithms.

We test our method on four different sequences containing varying motions. The first two sequences, *shoe* and *bears* contain non-rigid deformations but limited occlusions. The remaining two sequences contain points tracks from a *cars* sequence. The first, *car-segmented*, contains only point tracks from a moving vehicle. The second, *car-full*, contains tracks from the vehicle as well as the independently-moving background. Point tracks for each sequence were generated using our feature snapping technique. Column splitting in our measurement matrices is a result of feature loss and automatic reacquisition; we do not artificially split any tracks. We generate ground truth entries of the measurement matrix by hand tracking the acquired points, and we also determine ground truth associations for tracks. For the first two experiments, we assume known ranks. See Figure 2 for more details on the sequences.

6.1 Accuracy of Compaction

Table 1 shows our results on the compaction (data association) task. We compare our technique to two baseline association techniques: one that uses only geometric evidence and one that uses only photometric evidence. For the baseline appearance-only metric, we compute the photometric discrepancy matrix Φ and associate two tracks if they are respective best-matches ($i = \arg \min_j \phi_{ij}$ and $j = \arg \min_i \phi_{ij}$) with ϕ_{ij} below a threshold. We use our parameter λ_1 as the threshold. This method is limited as it cannot merge more than two columns. For the geometry-only method, we implement a version of the procedure described in Olsen and Bartoli [11]. We impute entries of the matrix using factorization without compaction, and then merge temporally consistent tracks with geometric discrepancy g_{ij} below a threshold corresponding to an average geometric error of one pixel per observation. We report error rates as the percentage of entries in C that are incorrect.

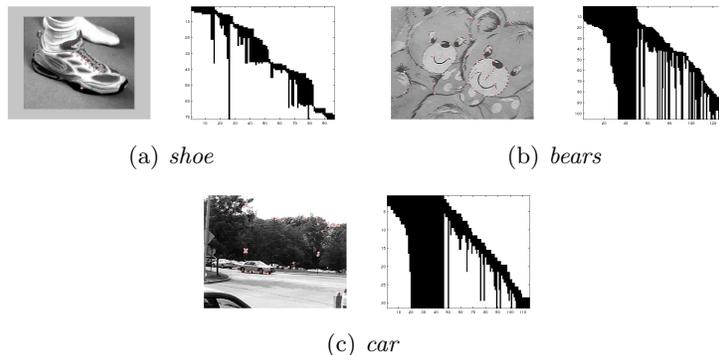


Fig. 2. Example frames and matrix fill patterns before compaction for the sequences we use. Tracking points through the 71 frames of the *shoe* sequence (a) generates 96 observations of 25 unique features. (M is 142×96 ; the true compact measurement matrix, \hat{M}^* , is 142×25 .) 85% of the entries in the measurement matrix are missing before compaction. Correct compaction reduces the percentage of missing data to 42%. The *bears* sequence (b) is the same as is used in Olsen and Bartoli [11]. It contains a total of 105 frames, and we track 69 unique features split into 131 columns. Before compaction, 62% of the entries in the measurement matrix are missing; after compaction only 28% are missing. The two cars sequences (c) use tracked points from the same 31-frame image sequence. In *car-full*, there are 115 columns from 60 unique tracks. We extract the 49 tracks that correspond to features on the car to form the *car-segmented* set.

Our algorithm recovers the true associations more successfully than the baseline alternatives. In fact, for the segmented car sequence, we recover the true associations exactly. For the *shoe* and *bears* sequences, our only errors are a few missed associations. Figure 3(a) shows an example missed association. We fail to group the track fragment on the right with the group containing the track on the left, likely due to the extreme deformation that occurs.

The worst association mistake we make is in the *car-full* sequence. Here, the tracks following the backs of the front and rear wheels of the car are each split into five separate track fragments. In the compaction matrix we recover, the last fragment from the back of the back wheel (left in Figure 3(b)) is merged with the group containing the tracks from the back of the front wheel (right). The temporal consistency constraint keeps us from merging the correct track fragment once this erroneous association is made.

6.2 Generalization Performance

Our ultimate goal is to be able to recover missing entries of the matrix M while simultaneously accurately reconstructing observed entries. The accuracy with which we can reconstruct observed entries is the *fitting error*:

$$\text{FE} = \sum_i \left\{ \frac{\sum_f F_2(f, i) (L(f, :)R(i, :)^T - M(f, i))^2}{\sum_f F_2(f, i)} \right\}. \quad (15)$$

Table 1. Accuracy of compaction, measured as the percent of the compaction matrix containing errors. Our technique returns a more accurate estimate of the track fragments to associate than relying on photometric or geometric information alone does.

Sequence	Percent of compaction matrix C incorrect		
	Photometric only	Geometric only	Compaction
shoe	3.27%	3.86%	0.13%
bears	0.79%	0.65%	0.09%
car-segmented	4.93%	5.27%	0.00%
car-full	0.95%	1.37%	0.24%

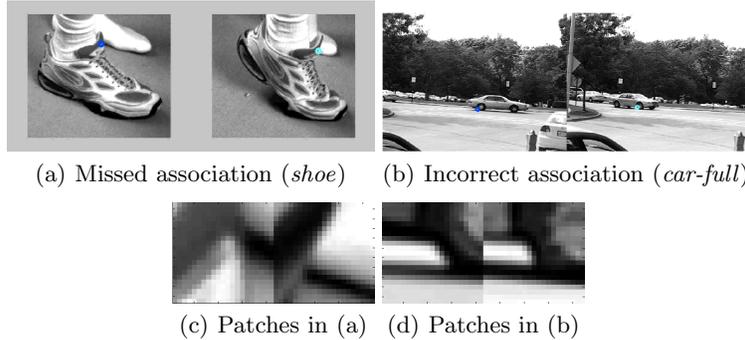


Fig. 3. Example of a missed association and an incorrect association during compaction. The full image pairs (a) and (b) show the location of track fragments in the first frame in which they are tracked. The remaining image pairs (c) and (d) show the local patches used to compute photometric discrepancy. In the *shoe* sequence, two track fragments are on the tongue of the shoe and should be merged but their photometric discrepancy is too great. In the *car-full* sequence, the local patches appear nearly identical although the tracks are actually on two different wheels. Decreasing the allowed rank from six to five eliminates this error but does not leave enough freedom in the reconstruction to accurately capture the full motion in the scene.

The fitting error is the same as the traditional RMS pixel error and is determined by how well the rank assumption is satisfied. In this work, we are uniquely focused on the *missing* elements of the measurement matrix so we measure reconstruction accuracy on the missing rather than the observed entries. The accuracy with which we can impute unknown entries of M is the *generalization error*:

$$\text{GE} = \sum_i \left\{ \frac{\sum_f (1 - F_2(f, i)) (L(f, :) R(i, :)^T - M^*(f, i))^2}{\sum_f (1 - F_2(f, i))} \right\} \quad (16)$$

where M^* contains the coordinates of the ground-truth tracks. Because we expect generalization to be easier over short distances, we consider generalization error as a function of the temporal distance to the nearest observation in M .

For example, if the last observation in the i th column of M was at time t , the missing entry at time $t + f$ has a temporal generalization distance of f .

We again compare our results to two baseline methods. The first demonstrates overall difficulty of the imputation task by using the true compaction matrix to form \hat{M} and then factoring (*true compaction*). The opposite extreme is a standard method that performs no compaction before factoring (*no compaction*). Generalization performance should be best for the true compaction method because the added geometric consistency constraint forces the reconstruction to be close to the observed location of a point any time it is observed, which is potentially many frames removed from the last observation of one of the corresponding fragments. Our results on the four sequences with known rank are summarized in Figure 4, with the performance of our algorithm (*estimated compaction*) shown in green. In every case, our technique significantly increases imputation accuracy over the standard approach.

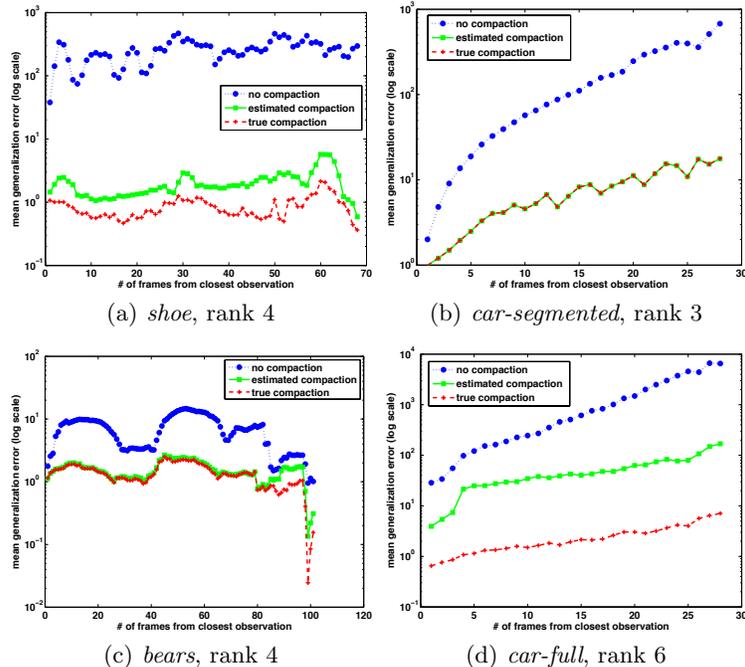


Fig. 4. Mean generalization error as a function of distance from nearest observation. Blue shows the baseline algorithm with no compaction. Red performs factorization after applying the ground-truth compaction. Our algorithm (green) estimates compaction and factorization simultaneously and results in improved performance over an algorithm without a compaction step. In (b), the red and green lines are identical because compaction recovers the true compaction matrix.

6.3 Performance with Incorrect Rank Estimation

Our last experiment investigates factorization performance when the rank of the true measurement matrix is overestimated. We shatter synthetic tracks into five track fragments, putting the first, third, and last fragment into individual columns and discarding the remaining two. Each original track is associated with a random patch from an image. We corrupt both the shattered measurement matrix and the local image patches with Gaussian noise. The true compact measurement matrix has 32% of the entries missing. After moving each track fragment to its own column, the percentage of missing data increases to 77%.

The true rank of (the noise-free) M is four. Figure 5 shows the generalization error with and without compaction for this sequence when factorization is allowed to use matrices of rank four, six, and eight. Without compaction, generalization error increases significantly with distance from the closest observation and with the rank of the reconstruction. Merely overestimating the rank by two increases the generalization error by an order of magnitude. With compaction applied, our algorithm is able to correctly impute even true missing entries of the measurement matrix (the discarded second and fourth segments of our original tracks, corresponding to generalization distances up to 30 frames) with errors on the order of the added geometric noise. More importantly, we maintain our accurate imputations as the rank is increased to six and then eight.

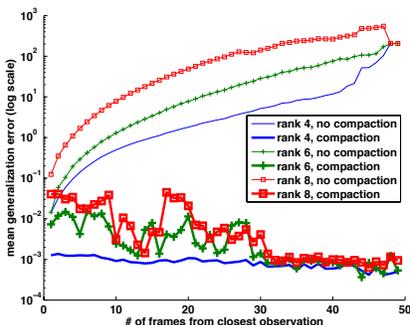


Fig. 5. Generalization error on a synthetic sequence of rank four using incorrect rank estimates. Without compaction (thin lines), generalization error increases rapidly with rank. With compaction (thick lines), performance is maintained even though the rank doubles. Generalization errors up to a distance of 30 frames correspond to imputations of true missing data. At distances beyond 30 frames, observations exist but have been incorrectly considered independent tracks in the shattered measurement matrix.

7 Conclusion

We have introduced a method for simultaneously compacting and factoring a matrix of measurements from feature tracking, as well as a feature snapping

technique that ensures precise alignment of those feature tracks that are merged during compaction. Preliminary experiments are promising. We successfully associate tracks across significant temporal gaps, and show improved imputation performance.

Our method is limited mainly by the local nature of the optimization algorithm and by the restriction of feature snapping to short-lived tracker failures. Our algorithm would not perform well if the tracker generates numerous outlier tracks as the interior factorization algorithm we used here, CSF, is not robust to outliers. We eliminated severe outliers by using conservative tracker parameters, relying on compaction to reconnect shattered track fragments. Also, while algorithm parameters can be determined in principled ways – mainly through cross-validation from training examples – we have not done so systematically in this paper. We plan to ease these limitations in future work.

Another limitation concerns the lack of extensive benchmark data for evaluating performance. We have shown results on several sequences from the literature, and we plan to continue developing our own validation database that we intend to share with the community.

Acknowledgements. This work is supported by the Army Research Office under Grant No. W911NF-10-1-0387 and by the National Science Foundation under Grant IIS-10-17017.

References

1. Bell, R.M., Koren, Y.: Lessons from the netflix prize challenge. *SIGKDD Explor. Newsl.* **9** (2007) 75–79
2. Tomasi, C., Kanade, T.: Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision* **9** (1992) 137–154
3. Costeira, J.P., Kanade, T.: A multibody factorization method for independently moving objects. *International Journal of Computer Vision* **29** (1998) 159–179
4. Bregler, C., Hertzmann, A., Biermann, H.: Recovering non-rigid 3d shape from image streams. In: 2000 IEEE Conference on Computer Vision and Pattern Recognition. Volume 2. (2000) 690 – 696
5. Brand, W.: Morphable 3d models from video. In: 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Volume 2. (2001) II–456 – II–463
6. Hartley, R., Schaffalitzky, F.: Powerfactorization: an approach to affine reconstruction with missing and uncertain data. In: Australia-Japan Advanced Workshop on Computer Vision. (2004)
7. Wiberg, T.: Computation of principal components when data are missing. In: 2nd Symp. Computational Statistics. (1976) 229–326
8. Buchanan, A., Fitzgibbon, A.: Damped newton algorithms for matrix factorization with missing data. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Volume 2. (2005) 316–322
9. Torresani, L., Hertzmann, A.: Automatic non-rigid 3d modeling from video. In: ECCV. Volume LNCS 3022. Springer (2004) 299–312

10. Gotardo, P.F.U., Martinez, A.M.: Non-rigid structure from motion with complementary rank-3 spaces. In: 2011 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. (2011) 3065–3072
11. Olsen, S., Bartoli, A.: Implicit non-rigid structure-from-motion with priors. *Journal of Mathematical Imaging and Vision* **31** (2008) 233–244
12. Del Bue, A.: A factorization approach to structure from motion with shape priors. In: 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. (2008) 1–8
13. De la Torre, F., Black, M.J.: Robust principal component analysis for computer vision. In: Eighth IEEE International Conference on Computer Vision. Volume 1. (2001) 362–369
14. Dellaert, F., Seitz, S.M., Thorpe, C.E., Thrun, S.: Structure from motion without correspondence. In: 2000 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Volume 2. (2000) 557–564
15. Triggs, B., McLauchlan, P., Hartley, R., Fitzgibbon, A.: Bundle adjustment – A modern synthesis. In Triggs, W., Zisserman, A., Szeliski, R., eds.: *Vision Algorithms: Theory and Practice*. LNCS. Springer Verlag (2000) 298–375
16. Shi, J., Tomasi, C.: Good features to track. In: 1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. (1994) 593–600
17. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: 7th International Joint Conference on Artificial Intelligence. (1981) 674–679