Damped Newton Algorithms for Matrix Factorization with Missing Data

A. M. Buchanan A. W. Fitzgibbon

Visual Geometry Group
Department of Engineering Science, Oxford University, UK
{amb,awf}@robots.ox.ac.uk

Abstract

The problem of low-rank matrix factorization in the presence of missing data has seen significant attention in recent computer vision research. The approach that dominates the literature is EM-like alternation of closed-form solutions for the two factors of the matrix. An obvious alternative is nonlinear optimization of both factors simultaneously, a strategy which has seen little published research. This paper provides a comprehensive comparison of the two strategies by evaluating previously published factorization algorithms as well as some second order methods not previously presented for this problem.

We conclude that, although alternation approaches can be very quick, their propensity to glacial convergence in narrow valleys of the cost function means that average-case performance is worse than second-order strategies. Further, we demonstrate the importance of two main observations: one, that schemes based on closed-form solutions alone are not suitable and that non-linear optimization strategies are faster, more accurate and provide more flexible frameworks for continued progress; and two, that basic objective functions are not adequate and that regularization priors must be incorporated, a process that is easier with nonlinear methods.

1. Introduction

Matrix factorization is central to many computer vision problems. Structure from motion (SFM) [12], illumination based reconstruction (IBR) [7], and non-rigid model tracking [3] all have solutions based in factorization. In each case, the measured data (interest points, pixel intensities) are observations of the elements of an $m \times n$ *measurement matrix*, denoted $\mathtt{M}_{\text{true}}$, which is of known[1] rank r, typically much smaller than $\min(m, n)$. With perfect data, in the absence of noise, the desired quantities are two smaller matrix *factors* $\mathtt{A} \in \mathbb{R}^{m\times r}$ and $\mathtt{B} \in \mathbb{R}^{n\times r}$ such that

$$\mathtt{M}_{\text{true}} = \mathtt{AB}^{\top}. \qquad (1)$$

In the presence of noise, $\mathtt{M}_{\text{true}}$ will not be observable, and we will instead observe a noise-corrupted version, $\mathtt{M}$. Assuming that the noise is isotropic Gaussian, the maximum likelihood estimates of $\mathtt{A}$ and $\mathtt{B}$ are the minimizers of the error function (with $\|\cdot\|_F$ denoting the Frobenius norm)

$$\epsilon_{\text{full}}(\mathtt{A}, \mathtt{B}) = \| \mathtt{M} - \mathtt{AB}^{\top} \|_F^2, \qquad (2)$$

for which there are reliable and globally convergent algorithms based on the singular value decomposition (SVD). In practice, however, an apparently innocuous modification of the problem must be solved, a modification that currently defies solution. The complication is that some elements of $\mathtt{M}$ may not be available, for example due to occlusions and tracking failures for SFM, or shadows and specularities for IBR. To account for these missing entries, a *weight* matrix, $\mathtt{W}$, is provided, of the same size as $\mathtt{M}$, in which zeros correspond to missing elements of $\mathtt{M}$. The modified factorization problem is now to compute the minimizers of

$$\epsilon_{\text{mle}}(\mathtt{A}, \mathtt{B}) = \| \mathtt{W} \odot \left(\mathtt{M} - \mathtt{AB}^{\top}\right) \|_F^2. \qquad (3)$$

where $\odot$ is the Hadamard product[2]. The search for an algorithm which can reliably minimize this error is the subject of this paper. Many approaches in the literature use an iterative strategy, which, although guaranteed to minimize the error at every iteration, is prone to *flatlining*: requiring excessive numbers of iterations before convergence (Figure 1).

Before a review of the literature and our contribution, a few more notes on (3). We observe that there is a *gauge freedom*: for any invertible $r \times r$ matrix $\mathtt{G}$, $\epsilon(\mathtt{A}, \mathtt{B}) = \epsilon(\mathtt{AG}, \mathtt{BG}^{-\top})$, meaning that for a minimizing pair of factors $(\mathtt{A}, \mathtt{B})$ there is an infinite family of equivalent solutions.

[1] In practice r is often unknown, but this paper shall assume it is given. There are strategies to choose r automatically, but in most real situations the user will be forced to choose either r or another tuning parameter which controls it.

[2] $\mathtt{R} = \mathtt{P} \odot \mathtt{Q} \Leftrightarrow r_{ij} = p_{ij} q_{ij}$

ALT	PF	SIR	HHH	BDT	AFAC	GA	
							inputs: $\mathtt{A}$,$\mathtt{B}$, $\mathtt{M}$, $\mathtt{W}$, r
1	1	1	1	1	1	1	1: $\hat{\mathtt{M}} \leftarrow \mathtt{M}$
2	2	2	2	2	2	2	2: **repeat**
						3	3: $\tilde{\mathtt{M}} \leftarrow$ truncate $\hat{\mathtt{M}}$ to rank r (via SVD)
4	4	4	4				4: $\mathbf{a}^i \leftarrow (\mathtt{B}^\top \mathrm{diag}(\mathbf{w}^i)^2 \mathtt{B} + \lambda_1 \mathtt{I})^{-1} \mathtt{B}^\top \mathrm{diag}(\mathbf{w}^i)^2 \hat{\mathbf{m}}_i \quad \forall i$
			5				5: $\hat{\mathtt{M}} \leftarrow$ update elements of $\hat{\mathtt{M}}$ with large residuals
				6	6		6: $\mathtt{A} \leftarrow$ first left singular vectors of $\hat{\mathtt{M}}$
		7	7				7: $[\ \mathtt{A} \quad \mathbf{t}\ ] \leftarrow \mathtt{A}$; $\hat{\mathtt{M}} \leftarrow \hat{\mathtt{M}} - \mathbf{t}\mathbf{1}^\top$
8	8	8			8		8: $\mathbf{b}^j \leftarrow (\mathtt{A}^\top \mathrm{diag}(\mathbf{w}_j)^2 \mathtt{A} + \lambda_2 \mathtt{I})^{-1} \mathtt{A}^\top \mathrm{diag}(\mathbf{w}_j)^2 \hat{\mathbf{m}}_j \quad \forall j$
	9						9: $\mathtt{B} \leftarrow$ column-wise orthonormalization of $\mathtt{B}$
				10			10: $\mathtt{B} \leftarrow$ Brandt closed-form update of $\mathtt{B}$
		11	11	11			11: $\mathtt{B} \leftarrow [\ \mathtt{B} \quad \mathbf{1}\ ]$; $\mathtt{A} \leftarrow [\ \mathtt{A} \quad \mathbf{t}\ ]$
		12	12				12: $\hat{\mathtt{M}} \leftarrow \hat{\mathtt{M}} + \mathbf{t}\mathbf{1}^\top$
					13		13: $\tilde{\mathtt{M}} \leftarrow (\mathtt{A}\mathtt{B}^\top)$
					14	14	14: $\hat{\mathtt{M}} \leftarrow \mathtt{W} \odot \mathtt{M} + (\mathtt{1} - \mathtt{W}) \odot \tilde{\mathtt{M}}$
15	15	15	15	15	15	15	15: **until** convergence
●	●	●	●	●	●		**outputs:** $\mathtt{A}$, $\mathtt{B}$
						●	**outputs:** $\tilde{\mathtt{M}}$

Algorithm 1. Alternation and variants for the minimization of Equation (4). Columns denote the lines used by each algorithm. ALT = alternation; PF = PowerFactorization [15]; SIR = Shum et al. [11]; HHH = Huynh et al. [8]; AFAC = Aanaes et al. [1]; GA = Guerreiro and Aguiar [6]; BDT = Brandt [2]. Notation: $\mathbf{x}^i$ is the i^{th} row and $\mathbf{x}_j$ the j^{th} column of matrix $\mathtt{X}$, with both $\mathbf{x}^i$ and $\mathbf{x}_j$ as column vectors.

Secondly, it is clear that not all weight matrices $\mathtt{W}$ admit a unique (even up to gauge) solution. As an extreme example, if $\mathtt{W} = \mathtt{0}$, all choices of $\mathtt{A}$ and $\mathtt{B}$ yield the same (zero) error. To cope with these and related problems, some of the algorithms we shall review minimize a modified version of (3):

$$\epsilon(\mathtt{A}, \mathtt{B}) = \|\mathtt{W} \odot \left(\mathtt{M} - \mathtt{A}\mathtt{B}^\top\right)\|_F^2 + \lambda_1 \|\mathtt{A}\|_F^2 + \lambda_2 \|\mathtt{B}\|_F^2, \quad (4)$$

with regularizing constants λ_1 and λ_2. Finally, a modification of (4) which is commonly encountered is the "PCA" or "SFM" modification, in which it is required that the r^{th} column of $\mathtt{B}$ is all ones, i.e. $b_{ir}{=}1\ \forall i$. All algorithms we propose are trivially modified to handle this case.

The remainder of this paper is as follows. We review the state of the art in minimizing (4) and remind the reader that the most effective current strategies are based around first-order minimizations, which we call "alternation". We then describe a class of second-order minimizations which converge more reliably than alternation and present experiments to support this claim.

2. Background

The literature on factorization with missing data falls into several categories: closed-form solutions, imputation, alternation, and direct nonlinear minimization of (4).

The **closed-form** solution introduced by Jacobs [9] does not minimize (4), but is the algebraically correct method to use. In Jacobs' algorithm, subsets of $\mathtt{M}$'s columns are used to build up the subspace orthogonal to $\mathtt{A}$. However, it is strongly affected by noise on the measurement data.

Imputation covers a number of strategies which attempt to fill in the missing entries of the matrix without considering the global error (4). For example, the original Tomasi and Kanade proposal [12]. Rather than tackle the whole problem at once, sub-blocks can be chosen to give a set of smaller and less sparse sub-problems. By making the sub-blocks overlap, the individual solutions may be 'stitched' together to give a solution for the whole problem.

In practice, using the closed form solution or imputation on real data, which tend to be noisy and have reduced coupling between known regions of the measurement matrix, produces poor results and so such methods are generally suitable only as initializations for the iterative algorithms. However, because we shall show that even the best iterative algorithms require multiple restarts from different starting points in order to find good optima, the single starting point these strategies supply is insufficient.

Alternation algorithms are based on the observation that if one of $\mathtt{A}$ or $\mathtt{B}$ are known, there is a closed-form solution for the other that minimizes (4). Algorithm 1 provides pseudo code that implements all of the alternation schemes proposed by the papers reviewed here and shows how all these algorithms are related. Lines 4 and 8 show the formulae for the closed-form minimizers of $\mathtt{A}$ and $\mathtt{B}$ respectively.

Wiberg [16] showed that alternation can be used to solve the factorization problem when data are missing. Later

Shum, Ikeuchi and Reddy [11] extended the algorithm to allow for arbitrary weighting values in W. It was shown by Roweis [10] that alternation can, in fact, be derived within an EM framework. Other variations on Wiberg's approach include Vidal and Hartley's suggestion [15] of adding a normalization step between the two factor updates. Also, Huynh, Hartley and Heyden [8] have proposed performing alternation on a continually updated version of M. Their algorithm does not include the weight matrix W, being aimed at outlier rejection problems. Aanaes et al [1], have put forward another method that works on an updated version of the measurement matrix. They use one alternation step after a subspace projection. Although not directly related, Guerreiro and Aguiar [6] also present a similar *project and merge* iteration scheme.

These algorithms have all been presented as solutions to the SFM/PCA problem. Shum et al's algorithm explicitly solves for the image centroid offsets, which are encoded in the last column of A, as seen in Algorithm 1. However, many do not deal with the fact that a measurement matrix with missing entries cannot be mean-centred. Brandt [2] has presented an algorithm, very similar to that of Aanaes et al, incorporating a closed form solution for B implicitly addressing this point.

A modification of (4) is minimized by de la Torre and Black [5], in which the Frobenius norm is replaced by a robust error function of the residuals. Their alternation updates are no longer closed form, but are implemented as a single step of a Newton-like algorithm which uses an approximation to the second derivatives of ϵ with respect to A and B separately. Thus, the comparison in this paper of full second-order methods, including the cross-derivatives $\partial^2\epsilon/\partial \mathtt{A}\partial \mathtt{B}$, to standard alternation, applies equally to this error function as well.

Torresani and Hertzmann [13] recast the estimation of the factors as Bayesian inference and, crucially, add priors on A and B, yielding excellent results on some difficult sequences. As we shall discuss in Section 5, the addition of priors greatly improves the veridicity of the recovered factors, but our conclusions regarding the best way to minimize (4) carry through to the with-priors case. The EM-based minimization in [13] produces updates which are closely analogous to alternation.

The final category of approach employs direct **nonlinear minimization** of the error function. Although not documented for the factorization problem, the use of second-order nonlinear optimization in projective structure and motion recovery problems has a long history [14]. Known as *bundle adjustment*, it depends typically on Levenberg-Marquardt (L-M) optimization of a Gauss-Newton approximation of (4). In the factorization case, the second derivatives are easy to compute (Appendix A), so we can employ a full Newton method. Section 3 describes the algorithm in more detail and the modification giving the damped Newton algorithm.

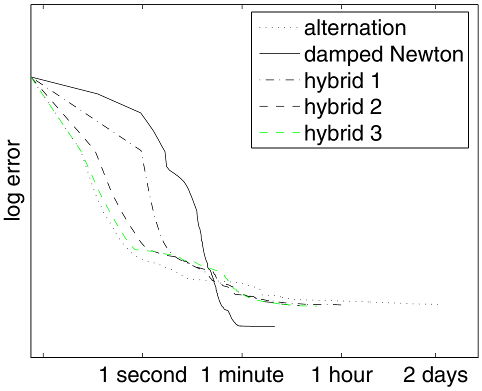


Figure 1. Typical error descent curves for the alternation, damped Newton and hybrid algorithms. All algorithms were started with the same random initial state vector. This is a minimization for the dinosaur sequence.

Summary: practical solutions to factorization and related problems fall into two main camps. Alternating closed-form solutions for each factor guarantees that the state error will be reduced after every iteration. In practice, the convergence of alternation iterations is initially very good, so combined with the attribute of typically fast iteration times, it is an attractive iteration scheme. However, we show that it is very susceptible to flatlining. This is because alternation is essentially a coordinate-descent scheme, albeit one in which the global optimum within each dimension subset is obtained at each descent step. Newton methods are expensive per iteration, initially converge slowly, and are fractionally more difficult to program, but on average require orders of magnitude fewer iterations to reach a solution.

In this paper we experimentally compare the two approaches (see [14] for a theoretical analysis). We also introduce a class of hybrid algorithms in an attempt to combine the advantages of both. Hybrid methods switch between alternation and Newton formulations as iterations progress, so that the periods of rapid convergence exhibited by alternation may be augmented with the long-term convergence power of damped Newton. This paper is, by necessity, concise; more details may be found in [4].

3. The Damped Newton Algorithm

In this section we vectorize the unknown variables, in this case the elements of the two matrix factors, so the error surface $\epsilon(\mathtt{A},\mathtt{B})$ becomes the function $\epsilon(\mathbf{x})$ of the state vector $\mathbf{x}$. At each iteration of the Newton method we seek an update $\delta\mathbf{x}$ which minimizes the second-order Taylor-series

```
inputs $\mathtt{A}, \mathtt{B}, \mathtt{M}, \mathtt{W}$
1:  declare F = $\left|\left|\mathtt{W}\odot\left(\mathtt{M}-\mathtt{A}\mathtt{B}^\top\right)\right|\right|_F^2+\lambda_1\left|\left|\mathtt{A}\right|\right|_F^2+\lambda_2\left|\left|\mathtt{B}\right|\right|_F^2$
2:  $\mathbf{x} \leftarrow$ vectorize(A,B)
3:  $\lambda \leftarrow 0.01$
4:  repeat
5:      $\mathbf{d} = \frac{\partial \mathrm{F}}{\partial \mathbf{x}}$
6:      $\mathtt{H} = \frac{\partial^2 \mathrm{F}}{\partial \mathbf{x}^2}$
7:      repeat
8:          $\lambda \leftarrow \lambda \times 10$
9:          $\mathbf{y} = \mathbf{x} - (\mathtt{H} + \lambda\mathtt{I})^{-1}\mathbf{d}$
10:     until $\mathrm{F}(\mathbf{y}) < \mathrm{F}(\mathbf{x})$
11:     $\mathbf{x} \leftarrow \mathbf{y}$
12:     $\lambda \leftarrow \lambda \div 10$
13: until convergence
outputs $\mathtt{A}, \mathtt{B} \leftarrow$ unvectorize($\mathbf{x}$)
```

Algorithm 2. The damped Newton algorithm

approximation

$$\epsilon(\mathbf{x}+\delta\mathbf{x}) \approx \epsilon(\mathbf{x}) + \frac{\partial \epsilon}{\partial \mathbf{x}} \cdot \delta\mathbf{x} + \frac{1}{2}(\frac{\partial^2 \epsilon}{\partial \mathbf{x}^2} \cdot \delta\mathbf{x}) \cdot \delta\mathbf{x}. \quad (5)$$

With the gradient $\mathbf{d} = \frac{\partial \epsilon}{\partial \mathbf{x}}$ and a positive definite Hessian matrix, $\mathtt{H} = \frac{\partial^2 \epsilon}{\partial \mathbf{x}^2}$, the minimum of this quadratic approximation is found using the update

$$\delta\mathbf{x} = -\mathtt{H}^{-1}\mathbf{d}. \quad (6)$$

Details of the computation of the gradient and Hessian are supplied in the appendix.

Unfortunately the error function in (4) is quartic in $\mathbf{x}$ and so quadratic approximations can be very unsuitable. Also, where the surface is almost flat in one or more dimensions the Hessian will be singular to machine precision. In these situations, when the $\mathtt{H}$ is not positive definite it is desirable to shift its eigenvalues. Adding a scaled identity matrix achieves this effect and results in a damped Newton method (Algorithm 2). It uses an adaptive regularizing parameter λ to perform this eigenvalue shift in an analogous way to the equivalent parameter in the L-M algorithm. The larger the value of λ, the smaller the length of the step taken by each iteration, reflecting the lower quality of the quadratic approximation, and resulting in more like gradient descent like behaviour.

Alternation-like Regularization. The damped Newton algorithm can be modified to resort to more alternation-like behaviour by boosting the diagonal $r \times r$ blocks of the Hessian using the λ parameter. In Matlab notation, replace $\mathtt{I}$ with `kron(eye(m+n), ones(r)).*H` on line 9 of Algorithm 2. Experiments consistently demonstrate that gradient descent is a very poor choice for minimizing (4) and so resorting to an alternation-style descent could be an advantage.

Damped Newton with Line Search. The matrix inversion on line 9 is the most expensive operation in the damped Newton algorithm and yet may be repeated several times per iteration. To reduce the number of inversions to one per iteration, determine the search direction using the Hessian, the gradient vector and λ, in the same way as damped Newton, but then use a line search to find the minimum in that direction. At the end of each iteration, λ is updated based on how far along the line the state was moved. If the step length taken was large, λ is reduced as the quadratic approximation was seemingly good. Conversely, a short step results in an increase in λ.

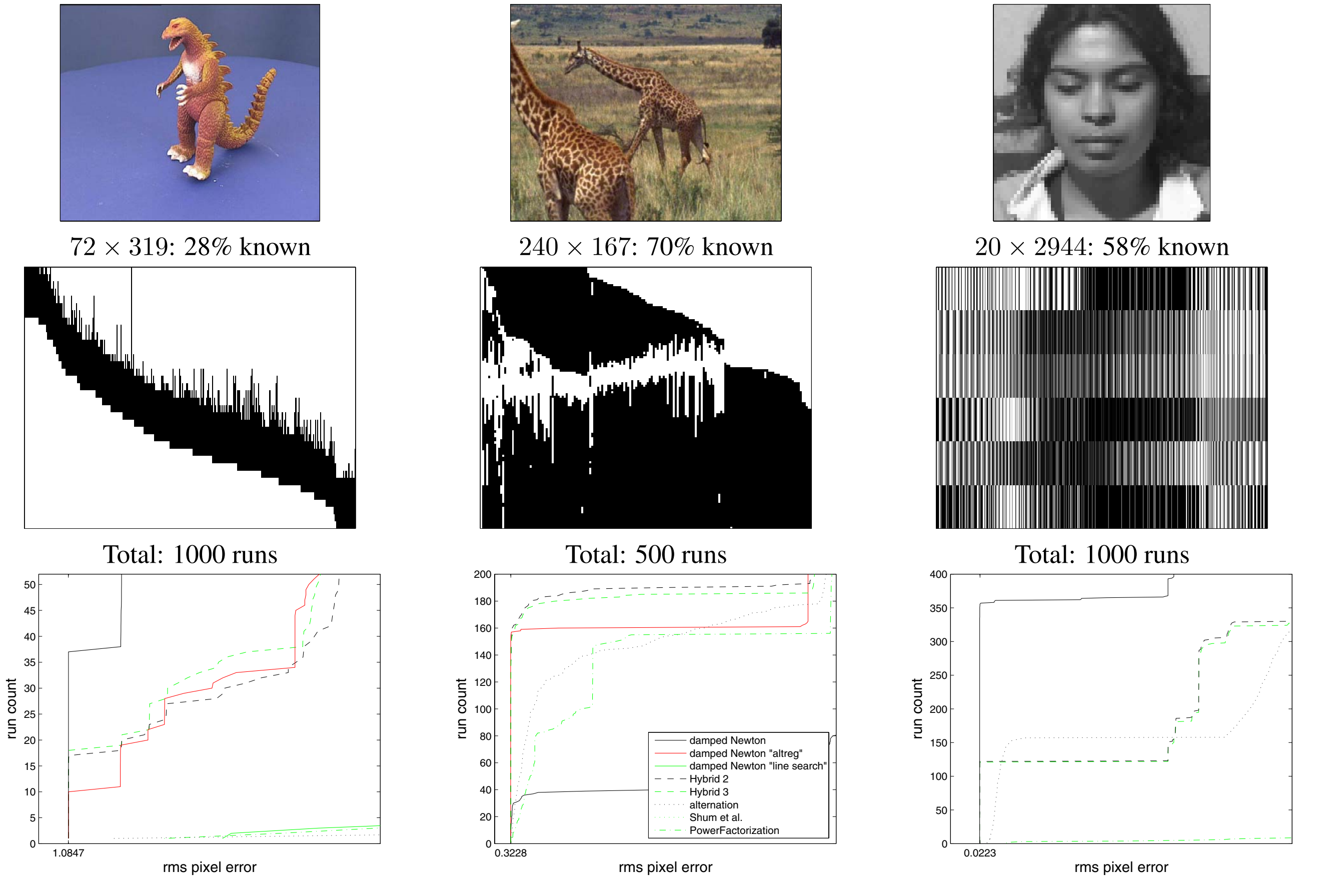
Hybrid Methods. A typical set of error descent curves are seen in Figure 1 giving a comparison of the alternation and damped Newton algorithms. Although other graphs could have been presented, the one shown represents the most common characteristics seen in experiments. Alternation initially converges very quickly, but slow convergence invariably dominates, typically taking thousands more iterations than the damped Newton algorithm. It would seem advantageous to combine the two methods to give an algorithm that has fast initial convergence giving way to the power of non-linear optimization. These are the hybrid schemes.

The crucial element of a hybrid algorithm is the switch criterion, i.e. how to make the decision to stop using alternation and start using damped Newton or vice-versa. We implemented three simple strategies: 1) calculate a new state with both alternation and damped Newton (with line search), then use the new state that gives the lowest new error; 2) run alternation for N iterations and switch to damped Newton. When λ rises to a threshold value, reduce λ and perform another N alternation steps; 3) use damped Newton with a check on λ within the inner loop. If λ exceeds a threshold value, perform an alternation step. The use of λ to initiate the switch comes again from the idea that it represents the fit of the quadratic surface approximation. Obviously, many variations on these basic ideas are possible.

Other Algorithms. Further to those tested, other derivative-based algorithms that might be compared with damped Newton are the conjugate gradient, quasi-Newton and limited memory quasi-Newton algorithms. Testing of these will be the subject of future work.

4. Experiments

All the algorithms reviewed and introduced above were run on three example problems, corresponding to the three applications mentioned in Section 1. The three problems also cover a range of matrix sizes and missing data ratios. A summary of the problems and results is given in Figure 2. Note that most of the tested algorithms cannot cope with outliers in the measurement matrix. Because of this, the



72 × 319: 28% known

240 × 167: 70% known

20 × 2944: 58% known

Total: 1000 runs

Total: 500 runs

Total: 1000 runs

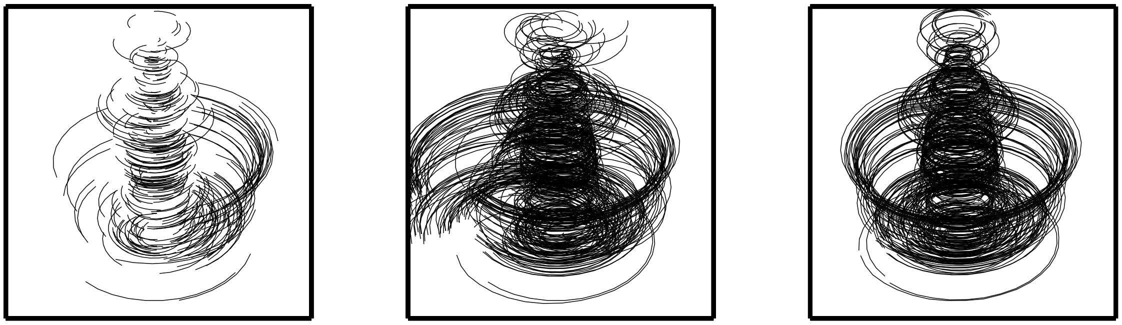
Figure 2. Summary of the results. The three columns correspond to three example problems: the recovery of 1) the rigid turntable motion of a toy dinosaur; 2) the non-rigid occluded motion of a giraffe in the background; and 3) the light directions and surface normals of a static face with a moving light source. The first row shows a frame from the sequence. The second represents the sparsity of the measurement matrix. The third row shows small sections of the accumulation histograms (curves showing the number of random initial state vectors which converged to a solution with a particular rms pixel error or lower) for the best of the algorithms presented in the text.

measurement matrix to be factored in each case was filtered to remove any outliers. This is discussed in section 5.

The algorithms were first initialized with potential factors given by the output of Jacobs's algorithm and secondly with a large number of random starting states (indicated in the figure). In all three cases the Jacobs estimate itself and the solutions given by the algorithms using Jacobs as a starting point were not as good as the best of the random runs.

It is assumed that accurate solutions are sought and poor solutions are to be rejected, irrespective of time taken. Therefore, timings will not be used as a measure of comparison. By this criterion several algorithms could be immediately discounted: the schemes proposed by Brandt, Huynh et al. and Aanaes et al., plus the project and merge algorithm never gave final errors below a modest threshold. The remaining three alternation algorithms (basic alternation, Shum et al. and PowerFactorization) all performed relatively evenly. However, on the dinosaur sequence, none of the pure alternation schemes achieved a final error as low as the best runs by the Newton-based algorithms. It must be noted that an iteration limit was imposed on the algorithms, and it is possible in some cases that a flatlined alternation algorithm could have reached the optimum. However, in our experience flatlining at 1000 iterations invariably mean that tens of thousands of iterations are required for convergence.

We confirmed that all runs that converged to the same error value gave the same factorization solution (within gauge freedom). Because the initial states were generated randomly, counting the number of runs that ended with the same error value is a reflection of the size of the basin of convergence for that minimum for each algorithm. The count for the lowest minimum can be taken as a measure of how well each algorithm performed on each example. When judged on this criterion, Newton-based methods outperform the other algorithms. The three alternation schemes performed with mixed success. Damped Newton itself was the most successful algorithm in the dinosaur and illumination problems and found the lowest error solution for the



~

rms pixel error = 1.0847

rms pixel error = 1.2702

Figure 3. Point tracks for the dinosaur turntable sequence. On the left are the input tracks. In the middle are the best tracks obtained using the algorithms discussed in Section 4. On the right are tracks obtained using simple priors (orthonormality of the camera matrices in A).

giraffe sequence as well. The hybrid schemes were the best algorithms considering performances across all three problems.

5. Discussion and Conclusions

Throughout the work above, it has been assumed that the error function (4) has a global minimum that represents a desirable and adequate result. Consider the dinosaur example. We are confident that the smallest error of all the 1000 runs is the global minimum for the dinosaur's measurement matrix. However, the solution associated with this minimum error is not satisfactory. Figure 3 shows how poorly the reconstructed tracks represent the turntable motion of this sequence. In an effort to obtain a better solution, extra terms were added to the error function (4) to penalize A having non camera-like properties (orthonormality) and damped Newton was run again. The third panel of Figure 3 illustrates the improvement obtained by including this simple prior. A very plausible three-dimensional reconstruction was also recovered. Note how the basic solution has a lower rms pixel error compared to the improved version, reflecting the fact that using models with more freedom will always fit noisy data better than constrained systems. Using the special-case solution as the initial state of the generic algorithms compliantly returns the inferior solution we labelled as the global minimum of (4). This reiterates the fact that for many real problems, it is not sufficient simply to minimize (4). Prior knowledge of the problem must be included in the minimization if good meaningful results are desired. As shown by Torresani and Hertzmann [13], the inclusion of priors can greatly improve performance on difficult problems. However, there are important consequences for the factorization problem: when priors are incorporated, the resulting error function frequently precludes the possibility of closed-form alternation. On the other hand, adding priors within a non-linear optimization framework is relatively easy. In theory, this will make alternation approaches even slower and emphasize the advantage of the damped Newton methods, but this claim remains to be tested. Alternation schemes implemented using non-linear forms are harder to create than pure Newton algorithms, but allow the construction of the hybrid form, which has been seen here to be successful in the context of basic error functions. Comparisons with more involved error functions is the subject of further research.

Also worthy of note is that the error surface of (4) is strongly affected by outliers. In the experiments for this paper, we removed outliers before performing factorization. This is not possible in general, and robust algorithms are required for reliable solutions. By including a robustifying term in the error function, as in [5], we again lose alternation's closed-form inner loop, and might again expect to see an advantage in second-order methods. This is also the subject of current investigation.

This paper has presented a comparison of many factorization algorithms. The comparison was made against Newton-based algorithms and hybrid schemes. The hybrid schemes performed very well within the context of the basic objective function (4), but updating them to use priors and robust error terms may be challenging. Newton-based algorithms themselves present an easy framework for such improvements. The results strongly suggest that second order non-linear optimization strategies are the key to successful matrix factorization when data are missing.

References

[1] H. Aanæs, R. Fisker, K. Åström, and J. M. Carstensen. Robust factorization. *PAMI*, 24(9):1215–25, 2002.

[2] S. Brandt. Closed-form solutions for affine reconstruction under missing data. In *Proceedings Stat. Methods for Video Processing (ECCV '02 Workshop)*, pages 109–14, 2002.

[3] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3D shape from image streams. In *Proceedings CVPR*, volume 2, pages 690–96, 2000.

[4] A. M. Buchanan. Investigation into matrix factorization when elements are unknown. Technical report, University of Oxford, `http://www.robots.ox.ac.uk/~amb`, 2004.

[5] F. de la Torre and M. Black. Robust principal component analysis for computer vision. In *Proceedings ICCV*, volume 1, pages 362–69, 2001.

[6] R. F. C. Guerreiro and P. M. Q. Aguiar. 3D structure from video streams with partially overlapping images. In *Proceedings ICIP*, volume 3, pages 897–900, 2002.

[7] H. Hayakawa. Photometric stereo under a light source with arbitrary motion. *JOSA*, 11(11):3079–89, 1994.

[8] D. Q. Huynh, R. Hartley, and A. Heyden. Outlier correction in image sequences for the affine camera. In *Proceedings ICCV*, volume 1, pages 585–90, 2003.

[9] D. W. Jacobs. Linear fitting with missing data for structure-from-motion. *CVIU*, 82(1):57–81, 2001.

[10] S. Roweis. EM algorithms for PCA and SPCA. In *Proceedings NIPS*, volume 10, pages 626–32, 1997.

[11] H. Shum, K. Ikeuchi, and R. Reddy. Principal component analysis with missing data and its application to polyhedral object modeling. *PAMI*, 17(9):855–67, 1995.

[12] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *Int. J. Computer Vision*, 9(2):137–54, 1992.

[13] L. Torresani and A. Hertzmann. Automatic non-rigid 3D modeling from video. In *Proceedings ECCV*, pages 299–312, 2004.

[14] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment – A modern synthesis. In *Vision Algorithms: Theory and Practice*, LNCS, pages 298–375. SV, 2000.

[15] R. Vidal and R. Hartley. Motion segmentation with missing data using PowerFactorization and GPCA. In *Proceedings CVPR*, volume 2, pages 310–16, 2004.

[16] T. Wiberg. Computation of principal components when data are missing. In *Proceedings Symposium of Comp. Stat.*, pages 229–326, 1976.

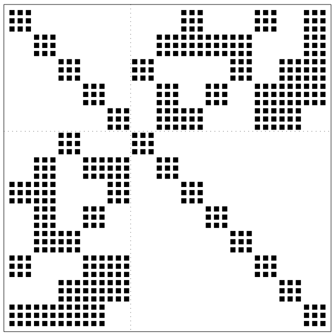


Figure 4. The sparsity structure of an example Hessian matrix. The upper left block is generated by Equation (14), the lower right block by (16) and the off-diagonal blocks by (15). This example was generated for $m = 5$, $n = 8$ and $r = 3$ with 50% known data.

A. Derivatives

The first and second derivatives of Equation (4) are required to implement Algorithm 2 and are given below. As in Algorithm 1, $\mathbf{x}^i$ is the i^{th} row and $\mathbf{x}_j$ the j^{th} column of matrix $\mathtt{X}$, with both $\mathbf{x}^i$ and $\mathbf{x}_j$ as column vectors.

A row-wise vectorization of $\mathtt{A}$ and $\mathtt{B}$, so that

$$\mathbf{x} = \begin{bmatrix} \mathbf{a}^{1\top} \; \mathbf{a}^{2\top} \; \cdots \; \mathbf{a}^{m\top} \; \mathbf{b}^{1\top} \; \cdots \; \mathbf{b}^{n\top} \end{bmatrix}^\top , \tag{7}$$

gives

$$\mathbf{d} = \frac{\partial \epsilon}{\partial \mathbf{x}} = \left[\frac{\partial \epsilon}{\partial \mathbf{a}^1}^\top \cdots \frac{\partial \epsilon}{\partial \mathbf{a}^m}^\top \; \frac{\partial \epsilon}{\partial \mathbf{b}^1}^\top \cdots \frac{\partial \epsilon}{\partial \mathbf{b}^n}^\top \right]^\top \tag{8}$$

$$\frac{\partial \epsilon}{\partial \mathbf{a}^i} = 2\mathtt{B}^\top \operatorname{diag}(\mathbf{w}^i)^2 \left(\mathtt{B}\mathbf{a}^i - \mathbf{m}^i\right) + 2\lambda_1 \mathbf{a}^i \tag{9}$$

$$\frac{\partial \epsilon}{\partial \mathbf{b}^j} = 2\mathtt{A}^\top \operatorname{diag}(\mathbf{w}_j)^2 \left(\mathtt{A}\mathbf{b}^j - \mathbf{m}_j\right) + 2\lambda_2 \mathbf{b}^j \tag{10}$$

$$\mathtt{H} = \frac{\partial^2 \epsilon}{\partial \mathbf{x}^2} = \left[\frac{\partial \mathbf{d}}{\partial \mathbf{a}^1} \cdots \frac{\partial \mathbf{d}}{\partial \mathbf{a}^m} \; \frac{\partial \mathbf{d}}{\partial \mathbf{b}^1} \cdots \frac{\partial \mathbf{d}}{\partial \mathbf{b}^n} \right]^\top \tag{11}$$

$$\frac{\partial \mathbf{d}}{\partial \mathbf{a}^i} = \left[\frac{\partial^2 \epsilon}{\partial \mathbf{a}^1 \partial \mathbf{a}^i}^\top \cdots \frac{\partial^2 \epsilon}{\partial \mathbf{a}^m \partial \mathbf{a}^i}^\top \; \frac{\partial^2 \epsilon}{\partial \mathbf{b}^1 \partial \mathbf{a}^i}^\top \cdots \frac{\partial^2 \epsilon}{\partial \mathbf{b}^n \partial \mathbf{a}^i}^\top \right]^\top \tag{12}$$

$$\frac{\partial \mathbf{d}}{\partial \mathbf{b}^j} = \left[\frac{\partial^2 \epsilon}{\partial \mathbf{a}^1 \partial \mathbf{b}^j}^\top \cdots \frac{\partial^2 \epsilon}{\partial \mathbf{a}^m \partial \mathbf{b}^j}^\top \; \frac{\partial^2 \epsilon}{\partial \mathbf{b}^1 \partial \mathbf{b}^j}^\top \cdots \frac{\partial^2 \epsilon}{\partial \mathbf{b}^n \partial \mathbf{b}^j}^\top \right]^\top \tag{13}$$

$$\frac{\partial^2 \epsilon}{\partial \mathbf{a}^p \partial \mathbf{a}^q} = \begin{cases} 2\mathtt{B}^\top \operatorname{diag}(\mathbf{w}^p)^2 \mathtt{B} + 2\lambda_1 \mathtt{I} & p = q \\ 0 & p \neq q \end{cases} \tag{14}$$

$$\frac{\partial^2 \epsilon}{\partial \mathbf{a}^p \partial \mathbf{b}^f} = 2 w_{pf}^2 \left(\mathbf{a}^p \mathbf{b}^{f\top} + \mathtt{I}(\mathbf{b}^{f\top} \mathbf{a}^p - m_{pf}) \right) \tag{15}$$

$$\frac{\partial^2 \epsilon}{\partial \mathbf{b}^e \partial \mathbf{b}^f} = \begin{cases} 2\mathtt{A}^\top \operatorname{diag}(\mathbf{w}_e)^2 \mathtt{A} + 2\lambda_2 \mathtt{I} & e = f \\ 0 & e \neq f \end{cases} \tag{16}$$