Iterative Projective Reconstruction from Multiple Views

Shyjan Mahamud Martial Hebert
Dept. of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

We propose an iterative method for the recovery of the projective structure and motion from multiple images. It has been recently noted [10, 13] that by scaling the measurement matrix by the true projective depths, *recovery of the structure and motion is possible by factorization. The reliable determination of the projective depths is crucial to the success of this approach. The previous approach recovers these projective depths using pairwise constraints among images. We first discuss a few important drawbacks with this approach. We then propose an iterative method where we simultaneously recover both the projective depths as well as the structure and motion that avoids some of these drawbacks by utilizing all of the available data uniformly. The new approach makes use of a subspace constraint on the projections of a 3D point onto an arbitrary number of images. The projective depths are readily determined by solving a generalized eigenvalue problem derived from the subspace constraint. We also formulate a dual subspace constraint on all the points in a given image, which can be used for verifying the projective geometry of a scene or object that was modeled. We prove the monotonic convergence of the iterative scheme to a local maximum. We show the robustness of the approach on both synthetic and real data despite large perspective distortions and varying initializations.*

1 Introduction

Many approaches exist in the literature for the recovery of structure and motion from a set of images of a rigid scene. One popular approach is the "stratified" approach [3, 4] where one first recovers structure and motion in a space that embeds the original euclidean space and then imposes appropriate metric constraints to recover the actual euclidean structure and motion. For example, assuming a perspective projection model for the camera, we can recover the projective structure and motion. Assuming calibrated cameras, we can then impose metric constraints to recover a euclidean reconstruction. If calibration parameters of the cameras are not known, one might be able to use self-calibration techniques for first recovering the calibration parameters of the camera before imposing metric constraints [3, 8]. The alternative is to simulatneously impose metric constraints during reconstruction assuming calibrated cameras (which we call the "direct" approach). A good critique of vaeious approaches is given by [7]. They note that if we restrict ourselves to the common case of an image sequence with fixed (perhaps with varying focal) and approximately known camera calibration, there might be no real advantage of using the "stratified" approach over other approaches as far as accuracy and performance is concerned. However, they do make the observation that in practice, the stratified approach tends to get stuck in local minima less often than the direct approach. We also feel that the "stratified" approach allows us to design hopefully simpler algorithms since we can treat each stage quite independently of the others.

When many images of the scene are available, ideally one should utilize all of the available data uniformly for recovering the structure and motion reliably. For approximate camera models like weak- or para-perspective, methods based on factorization have been presented in the literature [11]. Recently an elegant extension of this approach for the case of a projective camera model was suggested in [10, 13]. It is based on the insight that when the measurement matrix is scaled by the correct "projective depths", then the resulting scaled matrix has rank 4 and consequently the projective structure and motion can be recovered using a factorization based method. Their approach first recovers the projective depths from pair-wise constraints. While the factorization step does use all of the data uniformly, the procedure for recovering the projective depths does not treat the data uniformly, in order to keep the computations tractable. We propose an alternate iterative approach that recovers the projective depths simultaneously with the structure and motion and treats the data uniformly. In effect the new approach iteratively finds the projective depths that make the scaled measurement matrix globally "coherent".

In § 2, we review the technique of using the scaled measurement matrix for recovering the structure and motion. We then describe a few important drawbacks of the approach in [10, 13]. In § 4, we describe the new approach for iteratively recovering the projective depth along with the structure and motion while treating all of the available data uniformly. The approach utilizes a *subspace constraint* that is satisfied by all of the image projections corresponding to a 3D point. Solving a generalized eigen-problem resulting from this subspace constraint gives us the projective depths. We prove monotonic convergence to a local maximum for our iterative method. In § 3.1, a dual subspace constraint is presented for the projections of all the 3D points of the scene onto a given image. The dual constraint is useful for the run-time verification of the projective geometry of previously unseen images of the scene or object. In § 5 we report good results on both synthetic and real data despite large perspective distortions and varying initializations. We close with a disussion on extending the algorithm to handle missing data.

We have recently learned that our approach is similar in spirit to that of [1]. However, we note some important differences. The subspace constraint used there is dual to the one used here. More importantly, the objective function that the subspace constraints lead to are normalized differently. The normalization we use is arguably the more natural one but leads to a slightly more difficult optimization problem (requiring the solution of generalized eigenproblems), whereas the normalization in [1] is less intuitive but which makes the optimization more straightforward. Also, [1] does not provide a convergence proof (monotonic or not). Finally, we demonstrate the effectiveness of our method on scenes with significant perspective distortions compared with the ones reported in [1].

2 Scaled Measurement Matrix

Assume that we have m views of n 3D points. We wish to recover the projective structure of the n points along with the camera projection (or "motion") for each of the m views. Let P_i be the 3×4 projection matrix for image i, and let Q_j be the j'th 3D point of the scene in homogeneous coordinates. Then the projection equation for point j onto image i can be written as :

$$\lambda_{ij} q_{ij} = P_i Q_j \qquad (1)$$

where $q_{ij} = [x_{ij}, y_{ij}, 1]^T$ is the homogeneous image coordinate for point j in image i, and λ_{ij} is a scaling factor known as the "projective depth".

For m images and n points, let W be the $3m \times n$ measurement matrix $W = [q_{ij}]$. It is clear from equation 1 that the "scaled" measurement matrix $W_s = [\lambda_{ij} q_{ij}]$ has a rank 4 factorization $W_s = PQ$ where P is the $3m \times 4$ stack of projection matrices for all images (the "motion") and Q is the $4 \times n$ matrix of homogeneous coordinates for all points (the "structure"). Thus if we know the correct projective depths λ_{ij}'s, we can recover the structure and motion upto an unknown projective transformation [10, 13] using an SVD based factorization similar to that used in [11].

This approach is attractive since once the projective depths are known, all the data is utilized uniformly in recovering the structure and motion. However, recovering the projective depths reliably is crucial to the success of this approach. In [10, 13], they are recovered using pair-wise constraints among the projective depths. Briefly, there exists a pairwise constraint between the projective depths corresponding to the same point in two images through the fundamental matrix and epipoles of the two images. See [10, 13] for details.

There are a few important drawbacks to their approach for recovering the projective depths :

- The explicit recovery of the fundamental matrices and epipoles is required.
- Ideally, we would like to consider all the pair-wise constraints. However, for efficiency we can only consider a subset of all pair-wise constraints. The issue then is the choice of an appropriate subset of constraints. For images that come from a linear sequence, it might be sufficient to restrict ourselves to pairwise constraints of adjacent images. However with a sequence of constraints, errors in any given constraint can accumulate down the chain even if the rest of the constraints are accurate. For the more general case where the images need not come from a linear sequence, the choice of a good subset of pairwise constraints may not be obvious.
- The fundamental matrices and epipoles can only be recovered upto an unknown scale. If we wish to determine a reliable solution for the set of projective depths by using a redundant number of pairwise constraints, then we need a self-consistent scaling for the different fundamental matrices and epipoles. The suggestion in [10, 13] is to use quadratic identities among matching tensors, but this adds to the complexity of the approach.

We feel that the above approach for recovering the projective depths does not utilize all of the available data uniformly, thus diminishing the original attractiveness of the approach. We propose a simpler approach that iteratively finds the projective depths simultaneously with the structure and motion. Intuitively, we are searching for the projective depths that make the scaled measurement matrix globally "coherent", where coherence is measured by how well the the scaled matrix can be factorized into a rank 4 structure and motion. It will be seen that this "coherence" is attained precisely when the projections of each 3D point satisfy a a subspace constraint presented in the next section. In section § 4, we will use the deviations from the subspace constraints to iteratively recover the structure and motion.

3 The Subspace Constraint

The projections of a 3D point onto the m views satisfy a subspace constraint (this is related to the joint image of [12]). This section formulates the subspace constraint and shows that if the projective motion is known, then the projective depths for all the projections of a 3D point can be determined from the solution to an eigen-problem derived from the subspace constraint.

Consider the projection of a point Q_j onto the m images through the stack of projection matrices P :

$$s_j = [\lambda_{ij} q_{ij}] = P Q_j$$

where s_j is the column vector of the m homogeneous image coordinates scaled by the respective projective depths. Assume for now that P is known. It is clear from this equation that s_j lies in the subspace spanned by the columns of P with coefficients Q_j. Thus if we knew P, we can verify whether or not a set of image coordinates q_{ij} could have possibly come from the projection of some 3D point, if we can determine both a set of projective depths λ_{ij} and a projective point Q_j such that the column of scaled image coordinates s_j is spanned by the columns of P with combining coefficients given by Q_j. This is the *subspace constraint* that needs to be satisfied by any set of image projections that is assumed to be the projections of some 3D point. At this point, it might seem that apart from the knowledge of P one also needs to know the projective depths λ_{ij} before we can verify if a set of image coordinates satisfy the subspace constraint. As we shall see below, it is actually possible to verify the subspace constraint without first recovering λ_{ij} by solving for the largest eigenvalue for a corresponding eigen-problem.

The scaled image coordinates s_j lie on the subspace spanned by P if the residue of the projection of s_j onto P is zero :

$$R_j(\lambda_j) = \frac{|(PP^+ - I)s_j|^2}{|s_j|^2} = 0$$

where $\lambda_j = [\lambda_{1j} \cdots \lambda_{mj}]$ is the set of all m projective depths for the projections of point j, and P^+ is the pseudo-inverse of P. Note that we need to normalize the residue since otherwise we have the trivial solution $s_j = 0$ that we get from setting $\lambda_{ij} = 0$. We can simplify the expression for the residue above if we choose an orthonormal basis for the columns of P. Let U be a $3m \times 4$ matrix whose 4 columns are a set of some orthonormal basis that spans the columns of P. Due to the normalization, it can be verified that the above condition simplifies to the following condition (using the fact that $U^+ = U^T$ for the orthonormal basis U) :

$$G_j(\lambda_j) = 1 - R_j(\lambda_j) = \frac{s_j^T U U^T s_j}{s_j^T s_j} = 1$$

By separating out the unknown λ_{ij}'s from the known quantities q_{ij} and U, we have $s_j^T U = \lambda_j^T A_j$ where the i'th row of the $m \times 4$ matrix A_j is given by $q_{ij}^T U_i$ where U_i is the 3×4 matrix formed from the $(3i, 3i+1, 3i+2)$ triplet of rows of U. By also performing the same separation in the denominator of the ratio in the above condition we get :

$$G_j(\lambda_j) = \frac{\lambda^T A_j A_j^T \lambda}{\lambda^T B_j \lambda} = 1 \qquad (2)$$

where B_j is a diagonal matrix with the ith diagonal entry set to $q_{ij}^T q_{ij}$. We list the important properties of $G_j(\lambda_j)$ as a function of λ_j :

- Since the normalized residue $R_j(\lambda_j) = 1 - G(\lambda_j) \geq 0$, it follows that the maximum value of $G(\lambda_j)$ is 1.
- The maximum value of 1 is attained for some value of λ_j iff the set of image coordinates q_{ij} indeed corresponds to the projection of some projective 3D point onto the m views.
- Since $A_j A_j^T$ and B_j are symmetric matrices, a standard result from linear algebra [6] states that the maximum of $G_j(\lambda_j)$ is equal to the largest eigenvalue μ for the following generalized eigenvalue problem :

$$A_j A_j^T \lambda = \mu B_j \lambda \qquad (3)$$

Hence, to verify that a set of image coordinates q_{ij} corresponds to the projection of some 3D point j, we only need to verify that the largest eigenvalue of the above eigen-problem is 1 (neglecting noise). Specifically, we do **not** need to know the actual projective depths λ_j for the verification. Nevertheless we can recover the actual λ_j (as will be required in the iterative algorithm presented next) by solving for the eigenvector corresponding to the largest eigenvalue.

3.1 Dual Subspace Constraint

If we assume that we know the projective structure Q, there is a dual constraint on all the image coordinates from the same image i :

$$s_i = [\lambda_{ij} q_{ij}] = P_i Q$$

where s_i is the $3 \times n$ matrix whose column j contains the scaled image coordinate $\lambda_{ij} q_{ij}$, P_i is the projection matrix for image i and Q is the known $4 \times n$ structure matrix. In exactly the same manner as above (with different manipulation of the algebra), we can show that the dual subspace constraint is satisfied when the largest eigenvalue of a corresponding generalized eigen-problem is 1. The matrices of this eigen-problem are constructed from the image coordinates q_{ij} and the structure Q.

The dual subspace constraint is useful in cases where we need to verify that the projective structure of the points from a given test image are in fact generated from the recovered structure. Since we only need to determine if the largest eigenvalue is 1 (or close to 1 in case of noise), we note that in principle, we do not need to recover the projective depths explicitly as an intermediate step before the verification. Thus verification is in principle as direct as is the case for affine or euclidean structure. Nevertheless, since the residue is measured in projective space, it may not correspond very well to the actual metric error. For this reason, we will also have to compute the projective depths and compute the residue in the image plane of the reprojection of the projective structure.

4 Iterative Algorithm for Projective Structure and Motion

We now return to the problem of recovering the projective structure and motion. We require the projective depths λ_{ij} to construct the scaled measurement matrix $W_s = [\lambda_{ij} q_{ij}]$ which can then be factorized as $W_s = PQ$ to determine the projective structure Q and motion P. How do we determine the projective depths given only the image coordinates q_{ij} ?

We know that if we were also provided with P we could recover the projective depths λ_{ij} using the subspace constraint. Conversely, if we were provided with the projective depths, we can recover P by factorizing the scaled measurement matrix W_s. But neither P nor λ_{ij} are known. However, the circular dependence between P and λ_{ij} suggests the following iterative algorithm. Start with an initial guess for λ_{ij}, recover P from the scaled measurement matrix, then find the new λ_{ij} that satisfies the subspace constraint as "best" as possible and iterate till convergence. Here we list the important issues involved :

- What should be the initial values for λ_{ij} ? In most of our experiments, we set $\lambda_{ij} = 1$ which effectively means that we start with a weak-perspective approximation for the camera projection. However, we have confirmed that the performance of the algorithm is robust w.r.t. widely varying initializations. See § 5.

- Unless the iteration has converged to the right value for P, the current estimate P^k will not allow us to satisfy the subspace constraint exactly. Instead we should satisfy the subspace constraint as well as possible by finding the projective depths λ_{ij} and a projective point Q_j that will minimize the normalized residue between the vector of scaled image coordinates s_j and P^k. That is we should minimize :

$$R(\lambda_j, Q_j) = \frac{|U^k Q_j - s_j|^2}{|s_j|^2}$$

  where U^k is the orthonormal basis set spanning the columns of P^k. Note that we have to simultaneously solve for *both* Q_j and the λ_{ij} to minimize the above residue. It can be shown that the optimal values for Q_j takes the form : $Q_j = U^{kT} s_j^*$ where s_j^* is the optimal value for s_j [1].

- With the above observation, we can again show that due to the normalization, minimizing the residue $R(\lambda_j, Q_j)$ is the same as maximizing the objective function :

$$G_j^k(\lambda_j) = \frac{\lambda^T A_j^k A_j^{kT} \lambda}{\lambda^T B^k \lambda}$$

  where A^k, B^k are the same as in § 3, but constructed from the current estimate U^k.

The new projective depths at iteration k are determined by solving each of the eigen-problem $A_j^k A_j^{kT} \lambda = \mu B_j^k \lambda$ corresponding to each of the G_j's. These are

[1] This can be verified by imposing the necessary condition at the minimum : $\delta R / \delta Q_j |_{s_j = s_j^*} = 0$ and simplifying the resulting expression.

used to scale the measurement matrix whose rank 4 factorization gives us the next estimate U^{k+1} [2] for the orthonormal basis. There is one complication however : it will turn out that in order to guarantee the monotonic convergence of the algorithm, we also need to normalize each column j of the scaled matrix W_s^k. It can be verified from the projection equation 1 that normalizing a column j of the scaled matrix has the effect of changing the the scale of the corresponding homogenous point j. Arbitrary scaling of columns (and similarly of each triplet of rows) of the scaled matrix does not change the factorization of the matrix. These invariances to column-wise scalings are a degree of freedom that we exploit to gaurantee monotonic convergence (see the proof of the convergence theorem for details).

4.1 Convergence

Here we ask if we can say anything about the convergence of the algorithm presented in the previous section ? Will it always converge ? If so, will it converge to the global or a local maximum ? Will the convergence be strictly monotonic or on average ? We characterize the convergence property of the algorithm through an objective function whose maximum is 1 iff the subspace constraint is satisfied for the projections of all points in all views.

Definition 1 *Let λ be the set of all projective depths λ_{ij}. Define the following objective function :*

$$G^k(\lambda) = \frac{1}{n}\sum_{j=1}^{n} G_j^k(\lambda_j)$$

where the matrices A_j and B_j in G_j^k are constructed from the current estimate for the basis U. $G^k(\lambda) \leq 1$ since $G_j^k(\lambda) \leq 1$ for each j.

We note that $G^k(\lambda)$ attains the maximum of 1 (neglecting noise) for some value of λ iff the set of image coordinates corresponding to a 3D point j satisfy the subspace constraint for all j [3]. We now have the following convergence result.

Theorem 1 (Convergence) *Let λ^k be the value of λ at iteration k. Then $G^k(\lambda)$ converges monotonically*

$$G^{k+1}(\lambda^{k+1}) \geq G^k(\lambda^k)$$

to a local maximum.

Proof. See the appendix.

5 Experimental Results

5.1 Numerical considerations

As also noted in [10], we have found that the proper normalization of the image coordinates that was suggested in [5] is essential for good numerical conditioning of the factorization stage. For each image, all the image coordinates are translated so that they are centered at the origin of the image coordinate system and uniformly scaled so that the mean distance from the origin is $\sqrt{2}$. This does not affect the rank of the factorization since the above normalizing transformation, say T_i for image i simply transforms the corresponding projection matrix P_i to T_iP_i. Once the structure has been recovered, they can be un-normalized to recover the structure corresponding to the original coordinate system. Note that in addition to this normalization, the objective function being maximized implicitly also normalizes each column of the scaled measurement matrix at each iteration (see the proof of the convergence theorem). This is similar to the explicit normalization of each column in [10].

5.2 Evaluation Procedure

For each experiment, the available image data is divided into two sets : a "training" set from which we determine the structure and motion, and a "testing" set which is used to evaluate the accuracy of the recovered structure. It is important to evaluate the algorithm with a testing set that is different from the training set. It is not advisable to report the accuracy of the recovered structure on the training set itself since the algorithm could have "over-fitted" on the training set. Over-fitting is especially a concern for projective structure since there are more parameters to fit compared with the the actual underlying euclidean structure. For synthetic examples, we also report the errors after aligning the recovered structure with the 3D ground truth.

5.3 Synthetic examples

Many synthetic experiments were conducted to study the behavior of the algorithm w.r.t two issues : (a) the presence of large perspective distortions and varying noise, (b) varying initializations.

(a) Large perspective distortions : 30 points were selected at random within a sphere of radius 100 units. 10 training views of these points were taken from a camera whose origin was located at random points on a surface patch that was located at a distance of 150 units from the origin and projecting an angle of 30 degrees on the origin. The optical axis of the camera pointed towards the origin. The relatively small size of the patch on which the camera translates was chosen to make the task of recovery difficult since most

[2] Employing SVD for the factorization directly gives us U^{k+1} as the first 4 left singular vectors.

[3] Again use the fact that $G_j^k(\lambda) \leq 1$ for each j

of the information about the structure is embedded in the translations. Also, the relatively large size of the scene produces large perpective distortions. Note that since the camera positions were chosen anywhere on the 2D surface patch, there is no simple way to sequence the images that is required for applying the minimal number of pairwise constraints of [10]. An additional 10 views were taken from anywhere on a sphere of radius 150 units to serve as test images. Thus the test views were not confined to the small patch as the training views were. The parameters for the camera were (512, 512, 1). Varying gaussian noise levels of 0.0, 0.5, 1.0, 1.5, 2.0 pixels were added. All the statistics reported below are averaged over 20 trials with different initial seeds.

Figure 1 shows the pixel errors for a sample run where the pixel noise was set to 1.0. The range of the pixel errors are shown for each iteration. As can be seen from the range of pixel errors at iteration 0, the initial weak-perspective approximation (set with $\lambda = 1$) is quite poor. Nevertheless, by iteration 15, the average pixel error has dropped from around 20 to 0.7. Figure 2 shows the average 2D pixel error against the 10 test images after 20 iterations, while varying the noise level. By iteration 20 the algorithm converged despite the fact that the iterations start off with large average pixel errors in the range of around 15–40 pixels. Figure 3 shows the average 3D error after 20 iterations as a percentage of the scene width (the diameter of the sphere from which the points were sampled) after aligning the recovered projective structure with the 3D ground truth. None of the 20 trials converged to a local minimum (verified by measuring the distance of the minimum found to the true minimum from the ground truth).

(b) Varying initializations : In the previous experiments, the iterative algorithm was always started with the weak-perspective approximation, i.e., $\lambda_{ij} = 1$. We next explored the effect of random initializations where the initial value of **each** of the λ_{ij} was set to a random value in the range [0.5, 2.0]. A gaussian noise of 1.0 pixels was added. In 20 trials, none of the trials converged to a local minimum despite the fact that the average pixel error at the start of the iterations was in the wide range of 42–856 pixels for the varying initializations. The final error after 20 iterations averaged over 20 trials was 1.46 pixels.

5.4 Real Sequence

30 features were picked and manually tracked with ±1 pixel error across 20 images of a building. In one experiment (called "alternate"), alternate images in the sequence were used as training and testing data.

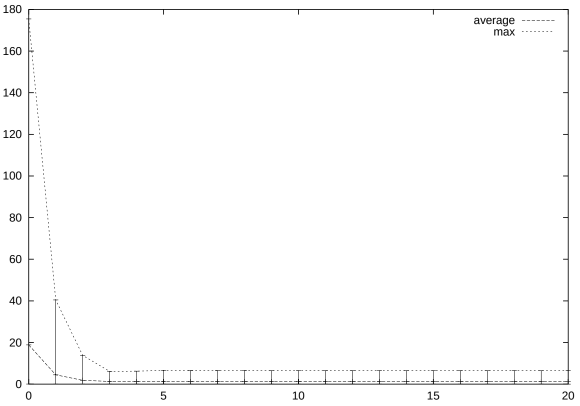

Figure 1: Sample run for a synthetic scene with noise set to 1.0 pixels. See text for the other parameters. Shown are the range of pixel errors over 10 test images vs iteration number. The average pixel error is around 0.7 pixels by iteration 15.

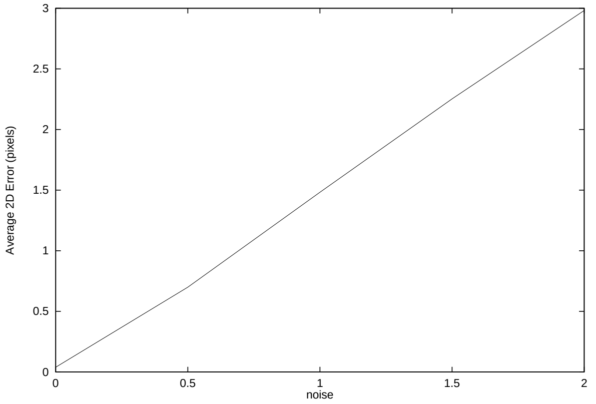

Figure 2: Average pixel error for the 10 test images of the synthetic scene after 20 iterations with varying noise levels. See text for details.

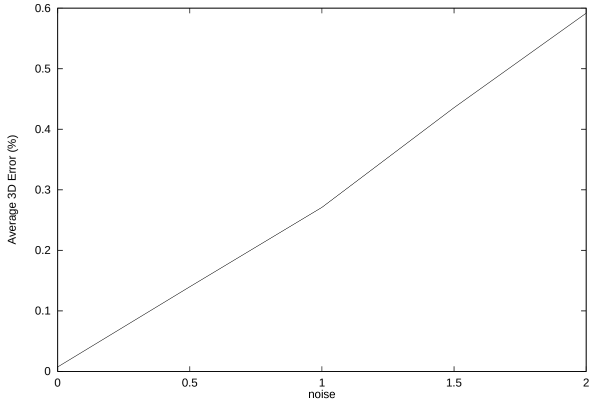

Figure 3: Average 3D error of the recovered structure of the synthetic scene after 20 iterations, after aligning the structure with the 3D ground truth.

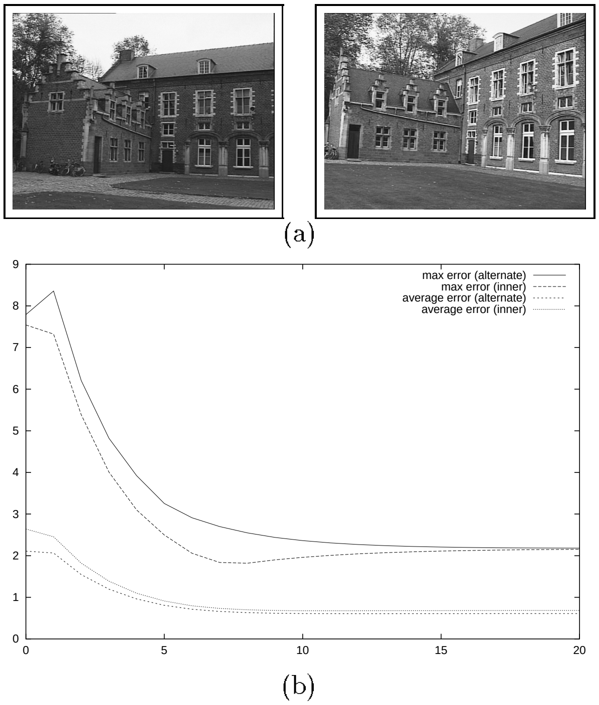

Figure 4: (a) First and last training images from the real image sequence. (b) Pixel errors for the separate set of test images. Shown are the range of pixel errors over 10 test images vs iteration number. The average pixel error is sub-pixel by iteration 15.

In a second experiment (called "inner") the middle 10 images of the sequence was used as the training set and the extreme 10 images were used as the testing set. The second experiment was designed to test the accuracy of the recovered structure despite restricting the overall range of translations of the camera.

Figure 4 shows the maximum as well as the average pixel error vs iteration number for both experiments. The maximum pixel error has dropped from around 8 pixels to around 2 pixels. The average pixel error converges to sub-pixel error in both experiments. Note that compared with the synthetic examples, the perspective distortions present in the scene are less for the real image sequence that we have.

6 Discussion

We have proved the monotonic convergence of the iterative algorithm to a local maximum. In [2], an iterative algorithm was presented for the case where camera calibration was assumed. No proof of convergence was presented there. It would be interesting to see if some of the analysis of our work can be applied to their algorithm. Alternatively, we can apply metric constraints to the projective structure recovered from our algorithm after taking into account the known calibration of the camera. The advantage is that we have a proof of convergence to a local maximum where there was none previously. This approach highlights one of the advantages of using a stratified approach to recovering structure and motion. Since each of the stages are usually simpler than the direct approach, they are hopefully more amenable to analysis.

In the future we plan to extend the algorithm to the case with missing data. As discussed in § 4.1, it is possible to modify the objective function for the projective depths to account for missing data. For all the image points corresponding to a 3D point j, we modify the corresponding term G_j by restricting the subspace considered to only that which is visible. Also the weight for each term should be proportional to the number of visible image points. One complication with missing data is that we can't resort to simple SVD-based factorization of the scaled matrix. Bilinear iterations are a possible candidate for factorization in the presence of missing data. These iterations will have to be done in a manner that will guarantee convergence.

Acknowledgements. The real image sequence is from K.U.Leuven

Appendix

Proof of the Convergence Theorem. At the start of iteration k, we have estimates λ^{k-1} and U^{k-1} from iteration $k-1$. During iteration k, we first fix U^{k-1} and maximize $G^{k-1}(\lambda)$ w.r.t. λ to get the new estimate λ^k. This implies $G^{k-1}(\lambda^k) \geq G^{k-1}(\lambda^{k-1})$, hence the value of G does not decrease. If it remains the same, we report convergence. If G increases we find the best 4-dimensional subspace U^k that spans a new measurement matrix W^k that is scaled by the new projective depths λ^k and whose column j is normalized by $(\lambda_j^{kT} B_j \lambda_j^k)$, i.e. :

$$W_{ij}^k = \lambda_{ij}^k q_{ij} / (\lambda_j^{kT} B_j \lambda_j^k)$$

Note that the rescaling of each column is a degree of freedom afforded by the task whose utilization is crucial for proving monotonic improvement of G. Once we form W^k, we compute the SVD of W^k and pick the left singular vectors U^k corresponding to the top 4 singular values. What remains to be shown is that $G^k(\lambda^k) \geq G^{k-1}(\lambda^k)$ which coupled with the above inequality implies $G^k(\lambda^k) \geq G^{k-1}(\lambda^{k-1})$ by transitivity and we are done.

To prove $G^k(\lambda^k) \geq G^{k-1}(\lambda^k)$, we use the *Poincare Extension Lemma* [6] which tells us the linear subspace that maximizes the projection of a matrix onto it (this is a generalization of the corresponding theorem for the well-known Rayleigh-Ritz ratio). The projection of a matrix W onto a subspace U is given by $\mathrm{tr}(U^T W W^T U)$. The Poincare Extension Lemma states that for a given r, the projection of W^k onto any r-dimensional orthogonal subspace U is maximum when the subspace is the first r left singular vectors U^* of W. More formally,

$$\max_{U^T U = I} \operatorname{tr}(U^T W W^T U) = \operatorname{tr}(U^{*T} W W^T U^*)$$

We can verify that the projection of W^k onto $U = U^{k-1}$ is nothing but $G^{k-1}(\lambda^k)$. Now let $r = 4$ in the Extension Lemma. U^{k-1} is some 4-dimensional orthogonal subspace possibly different from U^*. Also by design $U^k = U^*$ in our algorithm, whence $G^k(\lambda^k) \geq G^{k-1}(\lambda^k)$ as needed. Finally note that from the test for convergence, the derivative of G^k w.r.t. λ is 0 after convergence. Also from the Poincare Extension Lemma, U^k is chosen such that the derivative of G^k w.r.t. U is 0 at U^k. Hence, we have convergence to a local minimum.

References

[1] Berthilsson, R., Heyden, A. and Sparr, G., "Recursive Structure and Motion from Image Sequences using Shape and Depth Spaces", In *CVPR97*, pp 444–449, 1997.

[2] Christy, S. and Horaud, R. "Euclidean Shape and Motion from Multiple Perspective Views by Affine Iterations", *PAMI*, **18**(11):1098–1104, 1996.

[3] Faugeras, O. "Stratification of 3-Dimensional Vision: Projective, Affine, and Metric Representations", *JOSA-A*, **12**(3):465–484, 1995.

[4] Hartley, R.I. "Euclidean Reconstruction from Uncalibrated Views", In *CVPR94*, pp 908–912, 1994.

[5] Hartley, R.I. "In Defense of the Eight-Point Algorithm", *PAMI*, **19**(6):580–593, 1997.

[6] Horn, R.A. and Johnson, C.R. 1990. *Matrix Analysis*, Cambridge University Press, Cambridge, UK.

[7] Oliensis, J. and Govindu, V., "An Experimental Study of Projective Structure from Motion", *PAMI*, **21**(7):665–671, 1999.

[8] Pollefeys, M., Koch, R. and VanGool, L., "Self-Calibration and Metric Reconstruction in Spite of Varying and Unknown Internal Camera Parameters", In *ICCV98*, pp 90–95, 1998.

[9] Shashua, A. "Algebraic Functions For Recognition", *PAMI*, **17**(8):779–789, 1995.

[10] Sturm, P. and Triggs, B. "A Factorization Based Algorithm for Multi-Image Projective Structure and Motion", In *ECCV96*, pp II:709–720, 1996.

[11] Tomasi, C. and Kanade, T., "Shape and Motion from Image Streams under Orthography: A Factorization Method", *IJCV*, **9**(2):137–154, 1992.

[12] Triggs, B. "Matching Constraints and the Joint Image", In *ICCV95*, pp 338–343, 1995.

[13] Triggs, B. "Factorization Methods for Projective Structure and Motion", In *CVPR96*, 1996.