Optimal Multi-frame Correspondence with Assignment Tensors

R. Oliveira, R. Ferreira, and J.P. Costeira

Instituto de Sistemas e Robótica - Instituto Superior Técnico
{rco, ricardo, jpc}@isr.ist.utl.pt
http://www.isr.ist.utl.pt

Abstract. Establishing correspondence between features of a set of images has been a long-standing issue amongst the computer vision community. We propose a method that solves the multi-frame correspondence problem by imposing a rank constraint on the observed scene, i.e. rigidity is assumed. Since our algorithm is based solely on a geometrical (global) criterion, it does not suffer from issues usually associated to local methods, such as the aperture problem.

We model feature matching by introducing the *assignment tensor*, which allows *simultaneous* feature alignment for *all* images, thus providing a coherent solution to the calibrated multi-frame correspondence problem in a single step of linear complexity. Also, an iterative method is presented that is able to cope with the non-calibrated case. Moreover, our method is able to seamlessly reject a large number of outliers in every image, thus also handling occlusion in an integrated manner.

1 Introduction

The establishment of correspondence between image features extracted from different viewpoints of the same scene is an essential step to the the 3D reconstruction process. In fact, most reconstruction algorithms rely on previously established correspondences to determine 3D structure. Clear examples of this are classical factorization algorithms such as [15] and more recent methods as [6], [14] and [7]. A notable exception is presented in [3], where correspondences are not explicitly extracted - maximum likelihood structure and motion are calculated using an EM framework.

The difficulty of the correspondence problem is associated to its combinatorial nature. Furthermore, matching in multiple frames presents an additional difficulty to the traditional correspondence problem: coherence between every pairwise correspondence has to be guaranteed. Several models have been proposed in order to obtain a matching solution with an acceptable computational cost. In [12] and [8], the n-frame correspondence problem is formulated as a maximum-flow problem and is solved through graph cut algorithms. Different approaches involving graphs have been presented in [13] and in [5].

A natural way to associate a cost function to the correspondence problem is to exploit a constant characteristic of an important class of 3D scenes: rigidity. The

A. Leonardis, H. Bischof, and A. Pinz (Eds.): ECCV 2006, Part III, LNCS 3953, pp. 490–501, 2006.

use of rigidity presents the advantage of leading to intrinsically *global* algorithms; moreover, it naturally overcomes the aperture problem, since features are not characterized by their specific local properties. This geometric constraint can be translated into a rank constraint on the matrix containing the coordinates of the extracted features (the measurement matrix). Actually, it can be shown that when features in different viewpoints are correctly aligned (and only then) this matrix is highly rank-deficient - [9], [11]. Rank-deficiency for multi-frame correspondence has also been exploited in [10].

A first approach to correspondence exploiting rigidity has been made in [9], where the authors use a cost function based on the determinant of the measurement matrix to match features in a pair of images. This approach, although theoretically sound, has two main shortcomings: it is unable to handle the multi-image case and the cost function is intrinsically non-linear, presenting a high computational burden. In [11] the authors presented a new algorithm based on an alternative cost function, which would detect rank-deficiency based on the sum of the non-dominant singular values of the measurement matrix. This cost function allows the rigidity constraint to be applied to a multi-frame system. However, to obtain an acceptable computational complexity rank is imposed iteratively by matching each image individually with the remaining frames. Since rank is a global constraint this is not a desirable formulation. Moreover, occlusion cannot be modeled even within the iterative framework.

In this text, we propose a solution that generalizes the concept of assignment matrix used in our previous work to establish correspondences between the features in each of the frames. We introduce the assignment tensor that defines all correspondences in a single structure. With this formulation, linear complexity is retained even when dealing with more than two images, while occlusion is easily modeled.

2 Problem Formulation

We present in this paper a formulation that is capable of dealing with the multi-frame correspondence problem in the factorization context. Our objective is to align the observations in each image in a matrix W so that corresponding features share the same column. Optimal alignment is achieved by exploiting the intrinsic rank-deficiency associated to a correctly matched W.

Since our method relies solely on global geometric constraints of the scene, we place no constraints on the feature points selected - in particular, they do not have to contain significant texture in their vicinity. To emphasize this issue, our matching candidates are extracted from generic contour points, i.e. in areas prone to the aperture problem, and not from corners.

The method described herein assumes an orthographic camera, although it is easily extendable to any generic affine camera. In fact, the only factor limiting the camera model is the validity of the rank-defficiency condition on W.

2.1 The Assignment Tensor

Feature correspondence in a system containing n_f viewpoints is uniquely defined by a 2D point in each of the viewpoints such that all 2D points in the set are the projections of the same 3D feature. Bearing this in mind, it is straightforward to represent each correspondence in an n_f-dimensional structure - the *assignment tensor*. For the sake of simplicity, and without loss of generality, we will present these properties for the 3 image case. Extension to an arbitrary number of images is straightforward.

Suppose the three frames have p_1, p_2 and p_3 features, respectively. Each dimension of the assignment tensor contains a number of entries equal to the number of matching candidates (i.e. 2D features) in the associated frame: $i = 1, ..., p_1$, $j = 1, ..., p_2$, $k = 1, ..., p_3$. $T_{ijk} = 1$ iff the i^{th}, j^{th} and k^{th} features in the first, second and third frames respectively are projections of the same 3D point, i.e., if they represent a correct match. Otherwise, $T_{ijk} = 0$.

We represent the three-frame case as an example in Fig. 1 below.

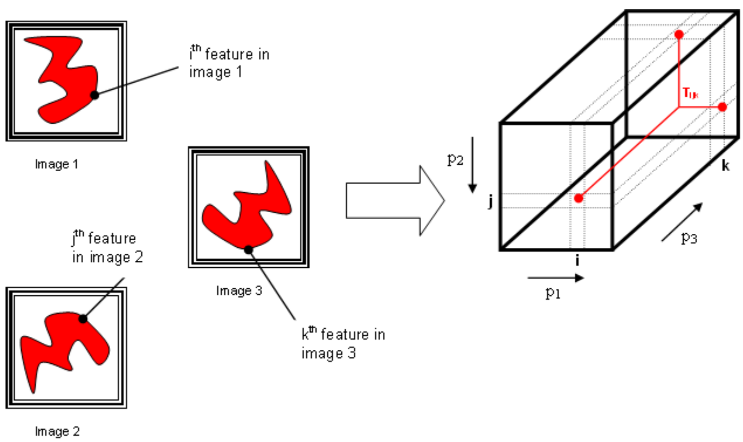

Fig. 1. The assignment tensor for the three-frame case. If feature i in the first image, j in the second and k in third are a valid correspondence, then $T_{ijk} = 1$.

Although the tensor establishes correspondence for all frames, the match between any subset of images can also be easily determined by summing over the dimensions not associated to the aforementioned images. In the 3-frame case, the relation P_{mn} between features in frame m and n can be easily obtained, as shown below (note that in this special case P_{mn} actually reduces to an assignment matrix). As will become evident in the next sections, the fact that any pairwise correspondence (represented by an *assignment matrix*) can be extracted from the assignment tensor is of key importance to our algorithm, as is the fact that the expression for each assignment matrix is linear in the elements of T. For the three image case, all pairwise correspondences are represented by the assignment matrices in Fig. 1:

$$P_{12} = \sum_{k=1}^{p_3} T_{ijk}, \quad P_{13} = \sum_{j=1}^{p_2} T_{ijk}, \quad P_{23} = \sum_{i=1}^{p_1} T_{ijk}. \tag{1}$$

To achieve a correct result, the assignment tensor must respect constraints that are intrinsic to the correspondence problem, such as unicity - a certain feature can be matched to *at most* one feature in another image. When matching a pair of images, this constraint is formulated by demanding that the sum of the rows/columns of the assignment matrix be less or equal to one. A similar set of constraints applies to the assignment tensor. In this case, it is required that *the sum over any dimension is less or equal to one*. This forces each feature to correspond to at most another feature in each of the remaining frames. For the three image case, the restrictions apply in the following manner:

$$\forall j,k \sum_{i=1}^{p_1} T_{ijk} \leq 1, \quad \forall i,k \sum_{j=1}^{p_2} T_{ijk} \leq 1, \quad \forall i,j \sum_{k=1}^{p_3} T_{ijk} \leq 1, \quad T_{ijk} \in \{0,1\}. \tag{2}$$

To avoid the trivial (and undesirable!) result of a null assignment tensor, a minimum number p_t of ones (i.e. matches) is forced on the tensor. This is done through the following restriction:

$$\sum_{i=1}^{p_1} \sum_{j=1}^{p_2} \sum_{k=1}^{p_3} T_{ijk} = p_t \tag{3}$$

The expressions for the three-frame case will be directly applied in the section dedicated to experiments.

2.2 Feature Point Representation

Observations on each frame are represented as a set of image coordinates containing the orthogonal projection of 3D feature points in the scene. Assuming p_f feature points, we represent the u and v image coordinates of a frame f in the u^f and v^f vectors. We assume that each set of p_f feature points is corrupted by a certain number of outliers which will have to be rejected. The data corresponding to frame f is thus represented by $2 \times p_f$ matrix w_f containing the u^f and v^f vectors.

Measurements corresponding to several frames can be vertically stacked in order to create a measurement matrix W_f that incorporates the projection of the feature points up to scene f. However, outliers in each frame have to be rejected beforehand; moreover, the remaining points have to be aligned so that corresponding features share the same column in W_f. Matrix P_f simultaneously aligns the feature points and rejects the outliers in the corresponding measurement matrix w_f. W_f can consequently be written as

$$W_f = \begin{bmatrix} w_1 \; P_1 \\ w_2 \; P_2 \\ \vdots \; \vdots \\ w_f \; P_f \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} u_1^1 \cdots u_{p_1}^1 \\ v_1^1 \cdots v_{p_1}^1 \end{bmatrix} & P_{1\,[p_1 \times p_1]} \\ \begin{bmatrix} u_1^2 \cdots u_{p_2}^2 \\ v_1^2 \cdots v_{p_2}^2 \end{bmatrix} & P_{2\,[p_2 \times p_1]} \\ \vdots & \vdots \\ \begin{bmatrix} u_1^f \cdots u_{p_f}^f \\ v_1^f \cdots v_{p_f}^f \end{bmatrix} & P_{f\,[p_f \times p_1]} \end{bmatrix} \quad (4)$$

We assume that only the best p_0 matches are to determined, where $p_0 \leq p_k, \forall k$. In (4), each P_k, for $k \geq 2$, represents a rank-p_0 assignment matrix which determines the correspondences between the first and the k^{th} frame. It has been seen in the previous section that these assignment matrices can easily be written as a linear expression of the terms of T as defined in (1). Under these assumptions, each assignment matrix is defined by the conditions in (5).

$$\begin{array}{ll} P_{k_{ij}} = \{0,1\}, \forall i = 1...p_k, \forall j = 1...p_0 & \sum_i P_{k_{ij}} \leq 1, \forall j = 1...p_1 \\ \sum_j P_{k_{ij}} \leq 1, \forall i = 1...p_k & \sum_{i,j} P_{k_{ij}} = p_0 \end{array} \quad (5)$$

Note that P_1 has a slightly different structure: it is a rank-p_0 matrix where ones are only allowed in the diagonal. Consequently, unlike the other P_k, it does not *permute* columns, it only forces certain columns of w_1 (corresponding to features that become occluded and thus do not have a match) to zero. As a result, W_f will have a set of null columns. This does not have any influence in subsequent calculations - in particular, this does not alter rank.

2.3 The Rank Constraint

It has been shown in [15] that a measurement matrix similar to the one presented in (2) is highly rank deficient. More specifically, when including translation W_f is at most rank-4. To this end it is however assumed that image coordinates corresponding to the same 3D feature point occupy the same column. In the presence of incorrect alignment, the resulting W_f is (generally) of higher rank. Note that in the presence of a limited amount of noise the rank-4 constraint for a correctly matched W_f may still be assumed as valid, as shown in [9].

Our problem is thus equivalent to *finding the correct assignment tensor* T. The tensor yields a set of assignment matrices P_k - each of these matrices aligns the corresponding w_k, so that a rank-4 W_f is generated.

2.4 The Cost Function

The multi-frame correspondence problem can be stated as the search for the assignment tensor that yields the optimal (pairwise) assignment matrices P_k as described in (1). The assignment matrices are optimal in a sense that these result in a rank-4 W_f (recall that W_f is a function of the assignment matrices). We consider the SVD decomposition of $W_f = Q\Sigma V^T$ and define Z as

$$Z = W_f W_f^T = \begin{bmatrix} w_1 P_1 P_1^T w_1^T & w_1 P_1 P_2^T w_2^T & \cdots & w_1 P_1 P_f^T w_f^T \\ w_2 P_2 P_1^T w_1^T & w_2 P_2 P_2^T w_2^T & \cdots & w_2 P_2 P_f^T w_f^T \\ \vdots & \vdots & & \vdots \\ w_f P_f P_1^T w_1^T & w_f P_f P_2^T w_2^T & \cdots & w_f P_f P_f^T w_f^T \end{bmatrix} \tag{6}$$

Recall that the aim of our algorithm is to find the matching solution that creates the best rank-4 W_f in the least-squares sense. This can be achieved by minimizing the sum of all eigenvalues λ_i of Z, with the exception of the four largest ones. This is a heuristic similar to the one used in [4], where rank minimization is achieved through minimization of the dual of the spectral norm. The eigenvalues of Z can be obtained, by definition, as the result of the following expression, where q_i represents the i^th column of Q, i. e. the i^th eigenvector of Z:

$$\lambda_i = q_i^T Z(P_1, P_2, ..., P_f) q_i, P_1 \in \mathcal{D}, P_2 \in \mathcal{P}^2, ..., P_f \in \mathcal{P}^f \tag{7}$$

where $\mathcal{P}^f$ represents the set of rank-p_0 assignment matrices of dimension $[p_f \times p_1]$ and $\mathcal{D}$ represents the set of rank-p_0 diagonal matrices of dimension $[p_1 \times p_1]$.The eigenvectors of Z are assumed known because these are the columns of Q, that under the factorization context is related to motion. In a calibrated system, Q is thus not a variable. For a rank-deficient Z, each of the non-dominant eigenvectors is a base vector for the null space of the column space of W_f defining in fact camera movement.

Our matching problem must thus be formalized as the search for the optimal set of assignment matrices $P_1^*, ..., P_f^*$ (e.g. optimal assignment tensor) such that:

$$P_1*, ..., P_f^* = \underset{P_1,...,P_f}{\arg\min} \left(\sum_{i>4} \lambda_i(P_1, ..., P_f) \right) = \underset{P_1...,P_f}{\arg\min} \left(\sum_{i>4} q_i^T Z(P_1, ..., P_f) q_i \right) \tag{8}$$

3 Minimizing the Cost Function Using Linear Programming

In general, solving the multi-frame correspondence problem through the minimization of (8) is a very tough problem. In particular, when considering only isolated assignment matrices, as was done in [11], the cost function in (8) is clearly quadratic, since there are certain terms (the crossed terms $w_i P_i {P_k}^T {w_k}^T, i \neq k$) which cannot be expressed as a linear function of the elements of the associated assignment matrices. Note that this is *not* an intrinsic property of the problem, but rather a consequence of an inadequate formulation: in fact, when working with single assignment matrices there are restrictions which are not considered. This is not the case with the assignment tensor, which takes into account all the inter-frame restrictions. In the present formulation, the crossed terms actually do have a linear form in the terms of the tensor - in other words, we can solve the correspondence problem as a linear problem. Moreover, we show that this problem can be easily solved through relaxation. In this section, the tensor formulation of the correspondence problem will be used to generate a linear formulation in the elements of T for the cost function presented in the previous section.

3.1 Unicity Constraints Revisited

The unicity constraints governing the structure of the assignment tensor, as they have been presented in (2), are awkward to use in the following calculations. We will consequently derive an equivalent formulation for these constraints.

We recall that the unicity condition requires that the sum over any dimension of the assignment tensor be *at most* one. Although these restrictions are trivially extendable to an arbitrary number of frames we will once more focus on the three-frame case, which allows a simple insight on the technique. This formulation would amount to:

$$\begin{aligned} \forall j,k,m,n, \sum_{i=1} T_{ijk}.T_{imn} &= \left(\sum_{i=1} T_{ijk}.T_{imn}\right).\delta_{jm}\delta_{kn}, \\ \forall i,k,l,n, \sum_{j=1} T_{ijk}.T_{ljn} &= \left(\sum_{j=1} T_{ijk}.T_{ljn}\right).\delta_{il}\delta_{kn}, \\ \forall i,j,l,m, \sum_{k=1} T_{ijk}.T_{lmk} &= \left(\sum_{k=1} T_{ijk}.T_{lmk}\right).\delta_{il}\delta_{jm}, \end{aligned} \tag{9}$$

In practice, this formulation is equivalent to saying that any two vectors in the same dimensions are orthogonal. This in turn will prevent two non-zero elements of the tensor of sharing the same dimension, thus enforcing the unicity conditions.

3.2 Solving for the Assignment Tensor

In this section we show that the cost function can be written as a linear program, thus effectively solving multi-frame correspondence with a low computational cost. Recall that the cost function has the following structure (the index i represents the order of the eigenvectors of Z):

$$\sum_{i>4} q_i^T W_f W_f^T q_i \tag{10}$$

Our objective is to extract the optimal assignment tensor T, which is uniquely determined by the optimal set of assignment matrices $P_2, ..., P_f$. Given a tensor T, the *vec* operator stacks its dimensions successively (from the first to the last) in order to form a vector: $x = vec(T)$.

Note the relation between T and the structure of the assignment matrices $(P_2, ..., P_f)$: the elements of these matrices are a linear function of the elements of T, as explained in section 2.1. Furthermore, products of matrices (such as $P_2 P_3^T$) actually represent pairwise correspondences (in this case between the second and third frame - P_{23}) and are thus also a linear cost function of x - the simplification becomes evident when using the constraints in the form presented in (9). Given each of the q_i, we can thus rearrange (10) as a linear function of x. Optimal correspondence will consequently be given by (11), where $\mathcal{T}$ represents the set of all assignment tensors of dimension $p_1 \times p_2 \times p_3$ and rank p_0; in the generic case, the number of dimensions contained in the dimension set is determined by the number of frames in the system.

$$x* = \arg\min_{x} c.x, \quad s.t.\ x = vec\left(T\right), T \in \mathcal{T} \tag{11}$$

The coefficient vector c can be calculated directly from the original formulation of the cost function by developing the expression in (10) in order to the elements of the assignment tensor. The calculation of c for the three image case is presented in the Appendix.

The formulation presented in (11) still remains an *integer* minimization problem and as such has no efficient solution. However, in the continuous domain there are algorithms that allow the solution to this problem to be obtained in a simple and swift manner. Fortunately, it can easily be shown that the assignment tensor possesses equivalent properties that allow an exact relaxation to take place - all that is needed is to demonstrate that the matrix containing the restrictions on the vector x is totally unimodular, as shown in [2].

The resulting problem is thus equivalent to the original, but for this class of problems (linear programming problems) there exist several efficient algorithms that can provide an adequate solution such as the simplex algorithm.

This method of solving the integer optimization problem has originally been proposed in [9].

4 Extensions to the Algorithm

4.1 The Non-calibrated Case

Up to this point, Q has been considered as known. However, an iterative solution has been devised which allows the solution of non-calibrated systems with small baselines provided a reasonably good initialization is available. Under this assumption, an initial estimate of the q_i is used to solve an approximate matching problem, which in turn returns an improved value for the q_i. This process is repeated until convergence is achieved; a similar method has already been published by the authors in [11].

Note that in the non-calibrated case two sets of unknowns are present: the elements of T and the columns of Q. Our iterative optimization scheme is analogous to a cyclic coordinate descent algorithm, in the sense that it optimizes a set of unknowns while keeping the remaining unknowns constant.

4.2 The Support Tensor

As can be inferred from the previous sections, the size of the linear program to be solved in order to obtain a matching solution can be potentially rather large. If some *a priori* knowledge is available, improbable matches can be excluded, thus reducing the dimensionality of the problem. To this end, a support tensor is used, which is a binary structure in which *allowable* matches are marked. All null variables are consequently eliminated from the x vector, thus rendering a smaller x^c vector.

5 Experiments

We describe in this section a set of experiments in order to validate the algorithm that has been presented. An experiment with real data provides a proof-of-concept solution, while demonstrating the ability of the algorithm to function under less than optimal conditions (i.e., with noise and deviations to the theoretical model). A non-calibrated example is also presented that illustrates how absence of information regarding motion may be circumvented.

5.1 The LEGO Grid

In this experiment three images of a LEGO grid are used. The grid defines two perpendicular planes in the 3D space. In this experiment, only contour points in the images are considered. In the first image, 99 points from the contour are selected as features. Note that the features are selected in areas where the contour is a straight line , so as to demonstrate the robustness of the method to the aperture problem. In the remaining images, the matching candidates are simply the contours of the images. In order to illustrate the handling of occlusion, parts of the contour have been removed in the second and third frames in order to create a situation under which some features in the first image do not have a valid match. No ground-truth is available, but correspondences can be verified by visual inspection.

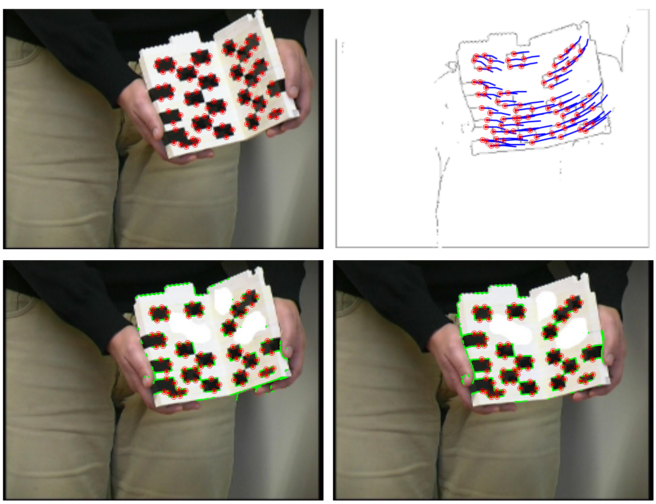

Fig. 2. Results for the LEGO Grid data set. Counter-clockwise from upper left: First image with features selected in red; second and third images with matching candidates in green and correspondences in red; feature trajectories in the third image.

Note that only a minimal error is noticeable by visual inspection, despite the fact that the camera was modeled as orthographic and that only approximate values were available for the q_i. Features which did not have valid matching candidates were successfully rejected. In this experiment a support tensor based on epipolar geometry was applied, so that candidates for each feature only exist in the vicinity of its epipolar lines. In total, ca. 3600 matching candidates were available for the 99 features in each of the frames. Using support, only 11000 matches were possible - consequently, *only a subset of the total number of matching candidates is an actual candidate for each feature*. It should be underlined that the use of the support tensor does not alter the result of the experiment; however, it does speed it up considerably - this problem, including support computation, can be solved in less than 15 min. on MATLAB. The actual matching algorithm, implemented in C, takes but a few seconds.

5.2 The Hotel Sequence

In this experiment information about camera motion is inexistent in the sequence, except in the first three frames. In the first two images 43 points (37 features and 6 points without matching candidates) have been singled out. Every image is matched against the first two using approximate values for motion information, i.e. the q_i. These are extrapolated based on the movement of frames already matched. These estimates are then iterated upon as referred in section 4.1. Note

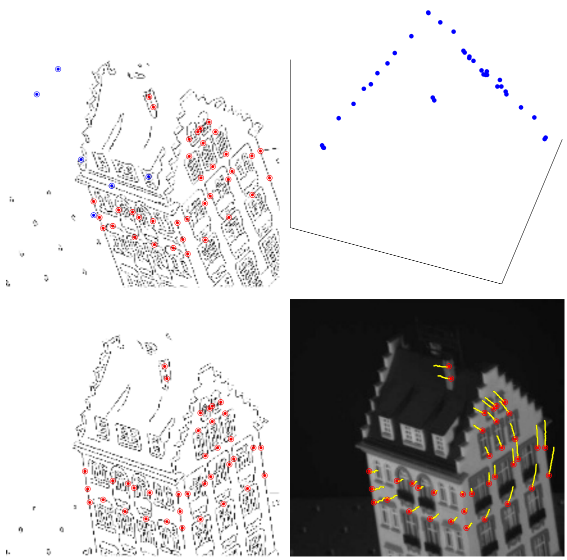

Fig. 3. Results for the Hotel data set. Counter-clockwise from upper left: First image with features in red and occluded points in blue; last image with correspondences in red; last image with trajectories; point cloud resulting from reconstruction, viewed from above.

that this is a simplified version of the presented algorithm, used only to illustrate the possibility of applying this work to uncalibrated images sequences; as such, matches are done pairwise to accelerate the procedure. Support based on maximum disparity between images is used.

No significant error is noticeable in this experiment, as the 37 features are correctly tracked and the 6 occluded points are rejected in every frame. Reconstruction based on the matches is precise. Each of the frames presents a total of ca. 11000 matching candidates, which after application of support reduces to only 1100 points.

6 Conclusions

We have presented in this text a novel approach to multi-view matching that allows correspondence to be obtained with linear complexity. This is achieved through a generalization of the concept of assignment matrix to the multidimensional assignment tensor. This tensor shares most of the properties of the assignment matrix, while adding constraints that allow a coherent solution between frames to be enforced. A cost function based on rigidity, as understood under the factorization context, has been used in conjunction with the assignment tensor to successfully determine correspondence between images. This cost function not only yields a global solution but also overcomes the aperture problem, owing to the fact that it does not depend on photometry as most present methods.

References

1. R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge Univerity Press 2000.
2. G. Nemhauser and L. Wolsey. *Integer and Cobinatorial Optimization*, John Wiley & Sons 1999.
3. F. Dellaert et al., Structure From Motion Without Correspondence. *In Proc. CVPR*, South Carolina, USA, June 2000
4. M. Fazel, H. Hindi and S. Boyd. A Rank Minimization Heuristic with Application to Minimum Order System Approximation. In *Proc. ACC*, June 2001.
5. V. Ferrari, T. Tuytelaars and L. van Gool. Wide-Baseline Multiple-View Correspondences. In *Proc. ICCV*, October 2003.
6. A. Heyden, R. Berthilsson and G. Sparr. An iterative factorization method for projective structure and motion from image sequences. *Image and Vision Computing*(17), 13(1), pp. 981-991, November 1999.
7. M. Irani and P. Anandan. Factorization with Uncertainty. In —textitProc. ECCV, June 2000.
8. V. Kolmogorov and R. Zabih. Multi-camera Scene Reconstruction via Graph Cuts. In *Proc. ECCV*, May 2002.
9. J. Maciel and J. Costeira. A Global Solution to Sparse Correspondence Problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25(2), February 2003.

10. D. Martinec and T. Pajdla. 3D Reconstruction by Fitting Low-Rank Matrices with Data. In *Proc. CVPR*, June 2005.
11. R. Oliveira, J. Costeira and J. Xavier. Optimal Point Correspondence through the Use of Rank Constraints. *In Proc. CVPR*, San Diego, USA, June 2005.
12. S. Roy and I. Cox. A Maximum-Flow Formulation of the N-Camera Stereo Correspondence Problem. In *Proc. ICCV*, January 1998.
13. K. Shafique and M. Shah. A Non-Iterative Greedy Algorithm for Multi-frame Point Correspondence. In *Proc. ICCV*, October 2003.
14. P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *Proc. ECCV*, pp. 709-720, April 1996.
15. C. Tomasi and T. Kanade. Shape from motion from image sreams under orthography: a factorization method. *IJCV*,9(2):137-154, November 1992.

Appendix

In this section an explicit expression for the coefficient vector of the linear program in (11) is presented, for the three-frame case. Each c_i is divided into a set of terms as in (12) corresponding, respectively, to the terms depending only on P_2, and P_3, and to the terms in $P_2{P_3}^T$, $P_1 P_1^T$, $P_2{P_2}^T$ and $P_3{P_3}^T$.

$$c_i = 2c_{P_2} + 2c_{P_3} + 2c_{P_2P_3} + c_P 1P1 + c_{P_2P_2} + c_{P_3P_3},$$

$$\begin{aligned}
c_{P_2} &= 1_{[1\times p_3]} \otimes \left({q_{i_{1:2}}}^T w_1 \otimes {q_{i_{3:4}}}^T w_2\right)\\
c_{P_3} &= \left({q_{i_{5:6}}}^T w_3 \otimes {q_{i_{1:2}}}^T w_1\right) \otimes 1_{[1\times p_2]}\\
c_{P_1P_1} &= 1_{[1\times p_3]} \otimes \left(\left(\left(q_{i_{1:2}}^T w_1\right) \bullet \left(q_{i_{1:2}}^T w_1\right)\right) \otimes 1_{[1\times p_2]}\right)\\
c_{P_2P_3} &= vec_r\left(\left({q_{i_{5:6}}}^T w_3\right) \otimes \left({q_{i_{3:4}}}^T w_2\right)\right)\\
c_{P_2P_2} &= vec\left(1_{[p_t\times p_3]} \otimes \left(diag\left(\left(I_{[p_2\times p_2]} \otimes {q_{i_{3:4}}}^T w_2\right) E_2{E_2}^T \left(I_{[p_2\times p_2]} \otimes {w_2}^T q_{i_{3:4}}\right)\right)\right)^T\right)\\
c_{P_3P_3} &= vec\left(\left(diag\left(\left(I_{[p_3\times p_3]} \otimes {q_{i_{5:6}}}^T w_3\right) E_3{E_3}^T \left(I_{[p_3\times p_3]} \otimes {w_3}^T q_{i_{5:6}}\right)\right)\right)^T \otimes 1_{[p_t\times p_2]}\right)
\end{aligned}$$

$$E_i = \left[e_1 e_1^T \cdots e_{p_i} e_{p_i}^T\right]^T \tag{12}$$

The vec_r operator acts in a similar way to vec, except that it stacks the rows of a matrix instead of its columns. e_i represents the i^{th} versor in the p_i-dimensional space. The complete c is constructed by the sum of all c_i.