Reduced Epipolar Cost for Accelerated Incremental SfM

A.L. Rodríguez
DITEC
Universidad de Murcia
alrl1@um.es

P.E. López-de-Teruel
DITEC
Universidad de Murcia
pedroe@um.es

A. Ruiz
DIS
Universidad de Murcia
aruiz@um.es

Abstract

We propose a reduced algebraic cost based on pairwise epipolar constraints for the iterative refinement of a multiple view 3D reconstruction. The aim is to accelerate the intermediate steps required when incrementally building a reconstruction from scratch. Though the proposed error is algebraic, careful input data normalization makes it a good approximation to the true geometric epipolar distance. Its minimization is significantly faster and obtains a geometric reprojection error very close to the optimum value, requiring very few iterations of final standard BA refinement. Smart usage of a reduced measurement matrix for each pair of views allows elimination of the variables corresponding to the 3D points prior to nonlinear optimization, subsequently reducing computation, memory usage, and considerably accelerating convergence. This approach has been tested in a wide range of real and synthetic problems, consistently obtaining significant robustness and convergence improvements even when starting from rough initial solutions. Its efficiency and scalability make it thus an ideal choice for incremental SfM in real-time tracking applications or scene modelling from large image databases.

1. Introduction

Bundle adjustment (BA) [28] is the standard postprocessing technique commonly used in computer vision to refine initial solutions obtained by different bootstrapping methods for the classical *structure from motion* (SfM) problem [14]. Currently, many efficient sparse bundle adjustment (SBA) software packages exist, based on well known nonlinear minimization techniques such as Levenberg-Marquardt (LM) [19] or Powell's Dog Leg [18]. These methods are able to refine a solution involving thousands of points and hundreds of cameras to the *maximum likelihood estimation* (MLE) in a well defined geometric sense, computing it in just a few seconds in a standard CPU. In fact, careful optimizations have recently allowed to even perform a significant amount of BA in real time visual mapping scenarios. This has been acknowledged to largely reduce failures in tracking [7, 15] in an arguably more robust way than strictly incremental approaches using Kalman state updates [6] or which limit the BA to the last two or three views [31]. Moreover, with the recent success of SfM for large scale scenes (such as extended areas of cities, for example) from huge unordered internet photo collections [24, 2], there has been a reborn interest in optimizing the BA process as much as possible [16, 1].

Anyway, BA requires an initial guess sufficiently close to the optimal solution. Otherwise, it can fall out of the convergence basin of the reprojection error. In realistic, medium/large scale SfM problems, direct linear initialization of all views and points at once [13, 11, 27] is problematic, so an incremental approach which builds up and refines the camera trajectory in sucessive steps is the de facto standard bootstraping procedure, both in sequential systems [8, 23, 7] and large scale object centered off-line reconstructions [25, 24, 2, 1]. These techniques tipically use BA in the growing subproblems to prevent divergence.

In this paper we propose to substitute the reprojection error in all intermediate BA stages for an alternative cost function based exclusively on the pairwise epipolar constraints. These constraints have been frequently used in the refining process of both structure and motion [31, 12]. In some cases, globally consistent epipolar (or even trifocal) constraints have also been used in a genuine multiview framework [29, 17, 26]. Our work is related to this approach, but, in contrast with it, we focus on the computational and qualitative advantages of a simplified algebraic cost for the full nonlinear optimization process:

1. Minimization is only performed for the camera parameters, completely discarding all variables corresponding to 3D points until final refinement, thus using much less variables in all intermediate optimizations.

2. It naturally allows to exploit the "second level" sparsity in the underlying linear system which arises from the fact that not all scene features appear in all views[1].

[1] This is related to the recent sparse-sparse bundle adjustment (SSBA)

3. In incremental bootstrapping scenarios it reuses many of its previous computations.

4. It shows remarkably fast convergence due to the regularization effect of the early elimination of structure variables in the optimization. Tipically only two or three iterations are required for stabilization. Furthermore, the LM parameter λ, which must be adaptive in standard BA, can be maintained fixed.

Refining approaches are sometimes classified as *unifying* vs. *decoupling* techniques [12]. Minimization of reprojection cost as in traditional SBA falls in the first class, trying to simultaneously optimize motion and structure. In contrast, we propose a decoupling approach based on a *global* epipolar cost function in which we first optimize only the motion parameters (i.e, position and orientation of the cameras), eliminating the 3D points. Triangulation is only performed at the end of the procedure, once the cameras have been fully refined, so our approach is essentially different from iterative intersection-resection [3].

The structure of the paper is as follows. In §2 we give a brief overview of bundle adjustment. Then §3 describes the proposed global epipolar cost function, while §4 emphasizes its computational advantages. In §5 we briefly discuss how to integrate this cost in a full incremental SfM system, and in §6 we experimentally study the convergence properties of the proposed minimization process, as well as its computational performance. Finally, in §7 we discuss the potential advantages, caveats, issues and possible extensions of the method, while §8 sumarizes the main conclusions of our study.

2. Sparse Bundle Adjustment

The maximum likelihood solution for structure and motion requires minimization of the geometric reprojection error of the observed image points. This is a high-dimensional, strongly nonlinear optimization problem, which can be successfully solved using the *Levenberg-Marquardt* (LM) method to iteratively refine an initial solution [28]. If $\vec{f}$ is the vector of residuals with Jacobian J, then the cost $C = \frac{1}{2}\vec{f}^T\vec{f}$ can be approximated by a quadratic function with gradient $\nabla C = J\vec{f}$ and Hessian $H \simeq J^T J$ (Gauss-Newton approximation). Therefore the correction Δx for the current solution is given by:

$$H\Delta x = -\nabla C \Rightarrow J^T J\Delta x = -J^T\vec{f} \tag{1}$$

The LM algorithm adaptively adds a scalar λ to the diagonal of the H matrix, in order to ensure that the cost function diminishes in each step even if the quadratic approximation is not valid [28, 7, 19].

General purpose LM implementations based on dense matrix algebra are not appropriate for SfM problems due to the high sparsity of the Jacobian, caused by the lack of interaction among parameters for different 3D points and cameras (each observed point depends only on a very small number of the total set of parameters). The standard approach to circumvent this problem is taking advantage of the block-diagonal structure of the Hessian associated to the points to reduce the full system of equations to a smaller one depending only on camera parameters using the Schur complement trick [1, 7].

In most practical situations points are seen only in small subsets of images, so the reduced camera system is also sparse [7]. Recent implementations like sSBA [16] exploit this second level of sparsity to achieve significant performance improvements. As it will be shown in the next sections, this will also be the case in our approach.

3. Global epipolar optimization

The global geometric epipolar cost can be defined as:

$$C_{ge} = \sum_p \sum_{i[p]} \sum_{j\neq i} d(x_i^{(p)}, E_{ij}\ \hat{x}_j^{(p)})^2 \tag{2}$$

where $d(x,l)$ is the Euclidean distance from a point x to a line l, E_{ij} is the Essential Matrix of the i-j view pair, $x_i^{(p)}$ is the observed image point p in view i, $\hat{x}_i^{(p)}$ is the estimated image of point p in view j, and $i[p]$ is the set of views i containing point p. (For simplicity we describe the fully calibrated situation, but the method could be easily extended to estimate also the internal parameters.) A simple approximation to this cost can be obtained if the epipolar lines arise from the observed $x_i^{(p)}$ instead of the estimated $\hat{x}_i^{(p)}$. In this case we can only include the epipolar lines induced by the available views $j[p]$. The empirical geometric epipolar cost is defined by:

$$C_{ege} = \sum_p \sum_{i[p]} \sum_{j[p]} d(x_i^{(p)}, E_{ij}\ x_j^{(p)})^2 \tag{3}$$

The approximation is good for realistic levels of noise and track lengths. This objective function can be further simplified if the epipolar distance is replaced by the algebraic cost of the constraint. Grouping the visible points $p[i,j]$ in each pair of views we can define the global algebraic epipolar cost as:

$$C_e = \sum_{i,j} \sum_{p[i,j]} \left(x_i^{(p)T} E_{ij}\ x_j^{(p)}\right)^2 \tag{4}$$

Since the method is used for refinement of an initial solution not very far from the optimum one, this algebraic cost

[16], but SSBA needs to previously eliminate structure variables using the Schur complement, while our proposed cost directly builds the sparse hessian matrix without ever using the 3D points.

can be made closer to the geometric one by means of adequate normalization and scaling of the input data, as discussed in the next section.

The proposed cost function has a particularly simple structure for Levenberg-Marquardt optimization. Each point p visible in views i and j induces a component in the cost function that can be expressed in terms of the rotations R_i, R_j and translations T_i, T_j of the cameras:

$$x_i^{(p)T} E_{ij}\, x_j^{(p)} = x_i^{(p)T} R_i \left[\frac{T_j - T_i}{||T_j - T_i||}\right]_\times R_j^T\, x_j^{(p)} \quad (5)$$

(Division by the norm of $T_j - T_i$ prevents convergence to the trivial solution $T_j = T_i$.)

The rotations can be parameterized by four elements of a quaternion, or by the three incremental Euler angles from the initial position. In any case, most of the computations required for the Jacobian of the above expression are very similar to the ones that are required for the camera parameters in standard BA. The Jacobian of the epipolar cost contains $O(Nl)$ nonzero blocks (or at most $N \times (N-1)$ in the fully connected case; see figure 1 left), where N is the number of cameras and l the average track length, and depends only on the camera parameters, while the Jacobian of SBA contains $N \times P \times 2$ blocks, where P is the mean number of points visible in each image. The 3D points do not appear in the optimization, though, if required, can be triangulated in a final stage from the optimal camera positions.

Notice that the H matrix corresponding to the proposed cost will have exactly the same sparsity pattern as the coefficient matrix of classical SBA after elimination of 3D points by the Schur complement trick: views not related by point correspondences will invariably produce zero blocks in the Hessian[2] (fig. 1 center, right). It is also important to note that the update equation (1) for the proposed algebraic error depends only on the camera parameters, so there is no need to perform any Schur complement to obtain a reduced matrix. In the next section we will see how to further reduce the computation, by *shrinking* the size of the jacobians.

4. Reduced measurement matrices

Each term $x_i^{(p)} E_{ij} x_j^{(p)}$ in equation 4 can be rewritten as the dot product of two $\mathbb{R}^9$ vectors, $m_{ij}^{(p)T} e_{ij}$, where $m_{ij}^{(p)} = x_i^{(p)} \otimes x_j^{(p)}$ and e_{ij} contains the elements of the matrix E_{ij}, normalized as in eq. 5. Equation 4 can then be rewritten in terms of the matrices M_{ij} whose rows are the vectors $m_{ij}^{(p)}$ corresponding to views i and j:

$$C_e = \sum_{i \neq j} \|M_{ij} e_{ij}\|^2 = \sum_{i \neq j} e_{ij}^T M_{ij}^T M_{ij} e_{ij} \quad (6)$$

We can obtain a numerically equivalent version of this expression by replacing the matrix M_{ij} (of size $n_{ij} \times 9$, where n_{ij} is the number of point correspondences between the views i and j) for a smaller upper triangular matrix $\hat{M}_{ij}$ of size 9×9 computed from the QR decomposition of M_{ij} (or, equivalently, from the Cholesky decomposition of $M_{ij}^T M_{ij}$), so that:

$$M_{ij}^T M_{ij} = \hat{M}_{ij}^T \hat{M}_{ij} \quad (7)$$

The *reduced measurement matrix* $\hat{M}_{ij}$ offers several appealing computational advantages against M_{ij}. First, its size is 9×9, independent of the number of point correspondences n_{ij}. Its memory footprint is much smaller (only 45 nonzero entries), and it largely decreases the computational cost of the components in the epipolar error function.

We precompute the reduced matrices $\hat{M}_{ij}$ before starting the optimization. In this stage the homogeneous calibrated observations $x_i^{(p)}$ are scaled to unit norm, and each row $m_{ij}^{(p)}$ in the original M_{ij} could optionally be weighted according to the initial E_{ij} to improve the geometric meaning of the cost. There are several ways to accomplish this effect [21], but our experimental results were not significantly better using this kind of additional prenormalization.

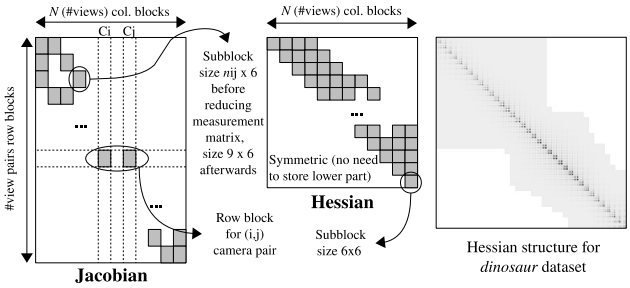

Figure 1. (Left) Sparse block structure of Jacobian and Hessian matrices for the proposed error function. (Right) Hessian matrix for the *dinosaur* dataset (§6).

4.1. Numerical Rank Properties

Besides the performance advantages, the reduced matrix also offers useful information for the reconstruction. Reasoning on its theoretical rank for different ideal, noise-free configurations of cameras and points, information about

[2]This sparsity can also be exploited by using a solver for sparse linear systems. In this work we have used the sparse LDL [9] decomposition in the Intel Math Kernel Libraries (MKL). Nevertheless, as will be discussed in §7, the iterative Preconditioned Conjugate Gradient (PCG) method [22] could be a better choice. The second level sparsity is not taken into account in some SBA implementations [7, 19], the iterative PCG approach is used in others [1], and some recent advances in direct Cholesky sparse linear systems [5] are used in [16] and also in [1]. In any case, due to the identical structure of the respective H matrices, any of these improvements could be used to equally speed up the linear equation solving step in our approach.

possible degenerate input configurations, presence of outliers and noise level magnitude can be inferred from the singular values of $\hat{M}_{ij}$. Let us ellaborate a bit on this. If we assume that the aforementioned unit norm scaling of each $x_i^{(p)}$ has been performed, then the following property always holds on the singular values of each $\hat{M}_{ij}$:

$$||\hat{M}_{ij}||_F^2 = ||M_{ij}||_F^2 = \sum_{k=1...9} s_k^2 = n_{ij} \quad \Leftrightarrow$$
$$||\bar{s}|| = 1, \quad \bar{s} = (\tfrac{s_1}{\sqrt{n_{ij}}}, \ldots, \tfrac{s_9}{\sqrt{n_{ij}}}) \tag{8}$$

Where $||.||_F$ stands for the Frobenius norm, s_k are the nine singular values of $\hat{M}_{ij}$ sorted in descending order, and $\bar{s}$ is the corresponding vector with module normalized to one. Given this, we can infer the following properties on the geometric configuration of this pair of views from the values of $\bar{s}_k = s_k/\sqrt{n_{ij}}$, assuming that $n_{ij} > 8$:

1. In a noise free general position of cameras and 3D points, the theoretical rank for $\hat{M}_{ij}$ is eight, so $\bar{s}_9$ should be zero. In a real situation, therefore, if this value is above a given *security* threshold, we can assume that there are outliers in the input matches.

2. Otherwise, $\bar{s}_9$ can still give us an idea of the noise level of the input, with values closer to zero giving us best view pairs (in terms of average projection noise).

3. For pure rotations (coincident camera centers), as well as for image pairs with all matching points coplanar in the 3D scene, there exists a homography relating the corresponding projections. In this case, the theoretical rank for $\hat{M}_{ij}$ is six, so $\bar{s}_7$ (and, of course, also $\bar{s}_8$ and $\bar{s}_9$) should be zero. In practical situations, thus, small values of $\bar{s}_7$ can be used to detect potentially ill conditioned pairs to get an initially correct (up to scale) 3D reconstruction from the estimated essential matrix.

4. Further zeroes on $\bar{s}_i$ for $i <= 6$ would indicate still more degenerate input configurations, such as all 3D points contained in a 3D line. This is perhaps less interesting from a practical point of view, as such situations should be rather infrequent in practice.

The above properties can be used as a guide to choose best image pair candidates to bootstrap a global reconstruction, much in the same spirit of [25]. In our case, however, we use information directly extracted from the corresponding reduced measurement matrices, instead of having to completely estimate the uncertainty associated to each pairwise reconstruction.

5. Application in a full SfM system

We have tested the proposed cost function in an incremental SfM system similar to the one described in [24] and used in [2] to successfully build large 3D reconstructions.

In the first step a standard keypoint matching algorithm is used to obtain point correspondences. At this point we also compute the reduced measurement matrix for each pair of views. This initialization task is evaluated only once, independently of the number of iterations required by the posterior optimization procedure.

The next step obtains an initial two-view reconstruction. We take advantage of the rank properties of the reduced measurement matrix to select a well conditioned view pair. We look for a pair with low $\bar{s}_9$ to ensure that the epipolar constraint is accurately satisfied, and with high $\bar{s}_7$ to avoid a degenerate geometry configuration due to planarity or small relative baseline. This makes unnecessary to estimate a planar homography between the image pairs as is suggested in [24]. Alternatively, an initial reconstruction with more than two cameras can be computed from the reduced measurement matrices using the technique described in [10].

Given a partial reconstruction, the algorithm iteratively resects new cameras, triangulates new points, and refines each augmented reconstruction using bundle adjustment to prevent divergence. In our implementation these intermediate refinement steps use the proposed epipolar cost to significantly accelerate the whole process.

A final refinement step using the standard reprojection error is performed to obtain the MLE solution. Since the bootstrap solution is very close to the optimum this last process requires very few LM iterations.

6. Experimental evaluation

In our experiments for testing the different optimization methods we used several datasets, each one containing a set of initial camera locations and 3D points (the latter not really being used by our proposal), and the corresponding image projections (measurements). The initial solutions were obtained by standard SfM techniques. Some databases correspond to real video sequences: *dinosaur*[3], *corridor*[4], and *maquette*[5] [19], and others are taken from large internet unordered photo collections [1][6].

6.1. GEA vs SBA convergence

We performed several tests to compare precision and convergence rate of epipolar and reprojection error. For these first group of tests we used our implementation (GEA,

[3]Thanks to Wolfgang Niem, University of Hannover.

[4]Oxford's VGG group, `http://www.robots.ox.ac.uk/~vgg/data/data-mview.html`.

[5]`http://www.ics.forth.gr/~lourakis/sba`

[6]`http://grail.cs.washington.edu/projects/bal/`

for *Global Epipolar Adjustment*), and the Lourakis and Argyros SBA package (laSBA) [19], perhaps the most tested freely available implementation for classical BA.

We evaluated the best reprojection error obtained using laSBA, GEA, and the number of iterations that each algorithm required to achieve it. We also evaluated the error and number of iterations obtained by laSBA when starting from the best GEA solution (label Both = GEA+SBA).

Table 1 summarizes the results. The stopping criteria used to consider that the algorithms have reached the optimal solution was to evaluate the relative decrement in the reprojection error obtained after each iteration. We show results for a typically used such decrement value, $\tau = 10^{-2}$. We corrected any camera distortions (linear, radial) from the measured point projections before evaluating the respective error values, so all the residuals are shown after being transformed to an euclidean frame (instead of pixel units).

	Reprojection error				Iterations		
Dataset	**Init.**	**SBA**	**GEA**	**Both**	**SBA**	**GEA**	**Both**
maquette	5.30e-4	3.49e-4	3.78e-4	3.62e-4	13	3	2
corridor	9.19e-3	1.08e-3	9.81e-4	9.37e-4	18	3	2
dinosaur	2.63e-2	4.01e-4	3.97e-4	3.91e-4	51	3	2
trafalgar-21	2.26e-3	7.07e-4	1.00e-3	7.16e-4	15	4	13
dubrovnik-16	3.53e-3	1.47e-3	1.62e-3	1.59e-3	19	3	2

Table 1. Optimal reprojection errors and convergence rates. The reconstruction provided by the dataset is used for initialization.

For a more comprehensive evaluation of the respective *convergence basins*, we additionally performed a second group of tests on artificially corrupted datasets. Starting from the optimal SBA solution of each problem, we corrupt the camera poses slightly, gradually varying their location and pose angle. We use then a linear algorithm to triangulate the 3D point locations with these new camera poses, and reevaluate the convergence rate (best reprojection error and number of iterations performed to reach it) of laSBA and GEA to their respective optimal solutions. Figure 2 show these results. Again, we also evaluate here the error and number of iterations obtained by laSBA, when initialized from the best GEA solution.

In general, the results of these convergence tests indicate that (*a*) GEA requires less iterations than laSBA to reach its optimal reprojection error, (*b*) a small number of SBA iterations suffice to obtain the best SBA solution starting from the GEA optimal solution, and (*c*) the *convergence basin* of SBA is significatively smaller than that of GEA, which is able to recover the correct solution starting from much larger initial reprojection errors[7].

[7]Notice that the *breaking point* of SBA, beyond which it will not be able to converge back to the correct solution, is always below ten times the optimal solution reprojection error, while GEA still converges to the correct solution from much more imprecise initial solutions (fig. 2 left).

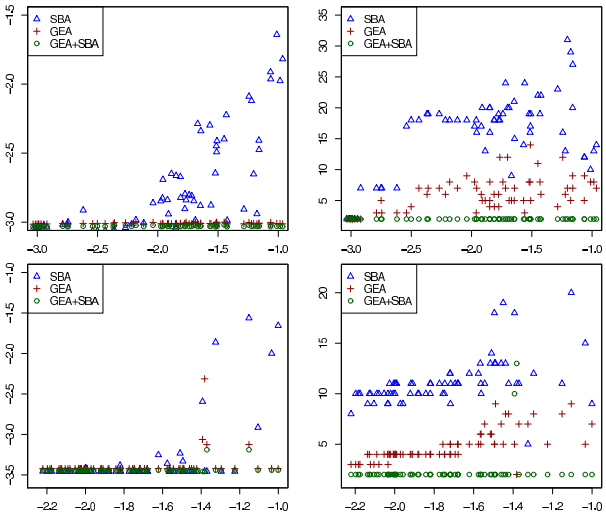

Figure 2. Convergence comparison between GEA and laSBA with artificially corrupted reconstructions. Vertical axis show best reprojection error obtained in logarithmic scale (left column), and the number of iterations required to reach it (right column), for the datasets *corridor* (upper row) and *maquette* (lower row). Horizontal axis represents the logarithm of the initial reprojection error.

6.2. Performance evaluation

We also compared the computational cost of our GEA implementation and, in this case, the sSBA package [16]. The sSBA is a recent BA implementation which offers state-of-the-art performance on large datasets, by solving the second level linear system using the CHOLMOD package [4]. Our GEA implementation uses the Intel Math Kernel Library (MKL) for that purpose. Both packages can use direct sparse Cholesky factorization as well as iterative PCG methods for solving the involved sparse system.

Table 2 summarizes the computing times per iteration obtained on the evaluated datasets for sSBA and GEA[8]. The columns labelled *Solve* correspond to the time spent in solving the second level system, using in this case the direct sparse methods from the MKL (GEA) and CHOLMOD (laSBA) respectively. The peformance of these solvers varies depending on the size and structure of the sparse matrices. MKL is faster when solving larger sparse systems, being as it is optimized for Intel architectures, but it tends to be slower in smaller systems, where it relatively spends more time initializing its internal data structures[9]. Columns labelled *Rest* include the rest of stages by iteration in both algorithms (compute respective cost functions, jacobians, and hessians, setup of coefficient matrix for the solver, and

[8]All tests were performed on an Intel Xeon 2Ghz single core.

[9]In any case, this is not the point here, as both GEA and sSBA would presumably consume approximately equal computing times in this stage when using the same direct solver implementation, because, by construction, both hessians have exactly the same sparsity structure.

update of the solution). Finally, columns *RM* and *LT* in GEA refer to initial reduced measurement matrices computation and final linear triangulation, respectively. Thus, the formula $T_{sSBA}(n) = n \times (Solve_{sSBA} + Rest_{sSBA})$ would model the time spent in practice by sSBA to optimize a given dataset on n iterations, while the corresponding time for n iterations of GEA would be $T_{GEA}(n) = RM + n \times (Solve_{GEA} + Rest_{GEA}) + LT$ (because computing of the reduced measurement matrices and the final linear triangulation of points must be performed only at the beginning and at the end of the GEA procedure, respectively).

To further clarify the computational advantages, Fig. 3 shows a detailed analysis of the number of iterations and overall computing times spent by both techniques vs. the achieved reprojection error in *dubrovnik-135*, an average size database. GEA obtains very good approximations to the optimum in much less time. Furthermore, the computing times for the proposed cost have been obtained with an unoptimized C++ prototype implementation, with priority on readability and ease of development instead of performance (unlike laSBA and sSBA, which are carefully implemented to exploit memory locality in the setup and computation of the involved matrices [19, 16]). In spite of this, for typical number of iterations needed by SBA and GEA (§6.1) we can see a clear advantage in favor of the latter.

	Problem size			sSBA		GEA			
Dataset	**#Cams**	**#Pts**	**#Projs**	**Rest**	**Solve**	**RM**	**Rest**	**Solve**	**LT**
trafalgar	50	20431	73967	223	6	67	12	20	296
trafalgar	126	40037	148328	516	79	158	40	101	588
trafalgar	256	65132	225700	797	997	282	105	331	957
dubrovnik	88	64298	383937	1771	29	746	47	88	1000
dubrovnik	135	90642	553336	3019	93	1341	104	216	1442
dubrovnik	356	226730	1255268	7024	4351	2977	479	1567	3442
ladybug	138	19878	85217	347	101	166	72	187	293
ladybug	460	56811	241877	1350	12250	639	458	1942	879
ladybug	1031	110968	500265	3372	34180	1437	982	6427	1718
venice	52	64053	347173	1438	7	645	18	24	976
venice	89	110973	562976	2415	30	1086	51	97	1654
venice	245	198739	1091386	6596	604	2837	285	738	3060
venice	427	310384	1699145	12387	14480	4708	736	2445	4871

Table 2. Performance times (milliseconds) of sSBA and GEA.

6.3. Robustness to varying λ in the LM algorithm

Finally, we evaluated the tolerance on the selection of the damping factor λ for both GEA and SBA algorithms. The sSBA implementation lets us specify this value, so we used it again in this last set of tests. To avoid eventual increase of the cost in some iteration, SBA must dinamically tune the λ parameter. In this case, we tested the influence of this parameter using both a direct and an iterative PCG solver for solving the sparse system of GEA and SBA, respectively. As shown in fig. 4, GEA convergence rate is independent of λ[10], while SBA shows a clear local minimum in both

[10]In practice we always use a small fixed value of $\lambda = 10^{-3}$, just to ensure positive definiteness of the coefficient matrix.

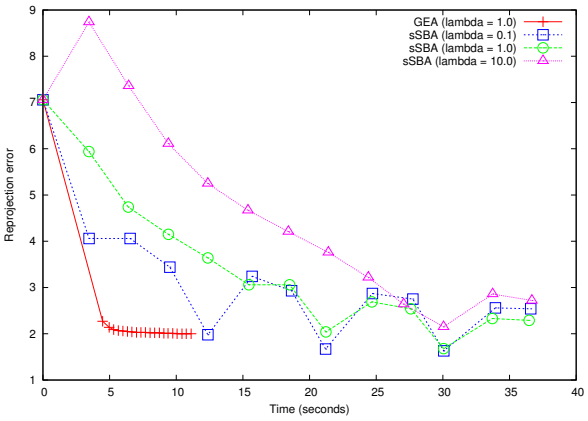

Figure 3. GEA and sSBA performance comparison for *dubrovnik 135*. The four curves correspond to GEA with $\lambda = 1.0$, and sSBA with three different initial values of $\lambda = 0.1$, 1.0 and 10.0. The marks in the curves correspond to projection errors achieved by increasing number of iterations in each method (time 0.0 corresponds in every case to initial projection error). We include in the first iteration of GEA the reduced matrices evaluation (performed at the beginning) and the posterior 3D point triangulation (which, though it is done at the end, is also performed only once). Observe how executing additional number of iterations in a GEA optimization does not significantly increase the overall computation time. Though not shown, convergence for GEA with a wide range of λ values is not significantly affected, while this value tends to influence much more in the convergence of sSBA, as can be appreciated in the three corresponding curves.

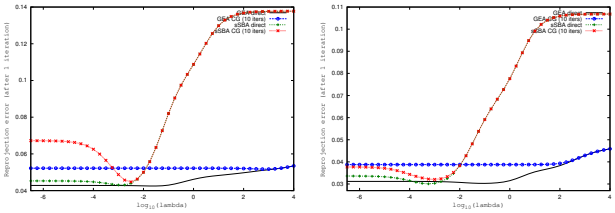

Figure 4. Reprojection error obtained after applying one iteration of GEA and the sSBA implementation, solving directly the second level system, or applying 10 conjugate gradient (CG) steps. The horizontal axis in these figures represents the initial λ value used in the iteration. The vertical axis represents the reprojection error obtained in one iteration. The datasets used are: *dubrovnik* with 16 cameras (left), and *trafalgar* with 21 cameras.

datasets and both types of solvers. This seems to corroborate the fact that the proposed cost behaves much more regularly (i.e., its quadratic approximation in the neighbourhood of the solution is better) than the SBA geometric reprojection error, which works on a much larger parameter space.

7. Discussion

In this section we discuss some important implications and further research lines raised by the proposed approach:

Alternative cost functions: The algebraic cost used in this work (eq. 4) is obviously not the only alternative. There are other more geometrically motivated costs, such as *symmetric transfer error* or the *Sampson error* [20, 30, 14, 21]. These simplified geometric costs would only slightly complicate the computation of the Jacobian, while still allowing early elimination of the 3D points in the optimization. In any case, our results with these costs show only marginal advantages with respect to the pure algebraic error, as already noted in the two views case[11] [14]. We discard the gold-standard epipolar geometric error (eq. 2), as it would need the 3D points in the optimization, thus throwing away the main computational advantage of the proposed cost.

Elimination of camera pairs: Another opportunity for optimization comes from the fact that the reduced matrices $\hat{M}_{i,j}$ corresponding to image pairs with a lot of matches will have much more impact in the global cost function than those corresponding to pairs with only a few. Nevertheless, every image pair contributes with the same weight to the global computational load (9 additional rows in the Jacobian). We have experimentally found that eliminating pairs with very few matches does not affect the obtained solution in an appreciable way. The analysis of the *views graph* to eliminate irrelevant image pairs from the Jacobian in intermediate stages of optimization in large problems will also be subject of future research. Since the Hessian has exactly the same block structure as the *adjacency matrix* of the *views graph*, this kind of optimization will also generate a more sparse system of equations (1), which could therefore be solved faster using the adequate sparse solving methods.

Reprojection error vs. epipolar cost in presence of point tracking failures: Minimizing reprojection error is the best alternative when we are sure that each track corresponds to a phisically different 3D point. But consider what happens when the tracker fails to detect an interest point for a few frames, generating two or more different tracks for the same 3D point (something frequent in real-time tracking systems). This provides additional degrees of freedom to BA to reduce the global reprojection error by artificial separation in space of the estimations of the common 3D point. Our experiments on synthetic data suggest that the epipolar cost is more stable against "track cuts" than standard BA, as only relative positions of the views induced by the matches (and no physical 3D points) are used in the optimization, showing a benign regularization effect.

Sufficient parallax: As already noted in [21], we have experimentally checked that even for purely rotational motions, the spurious artificial translation created by image noise is sufficient for correctly estimating the relative rotation between views, while the translation automatically converges to a very small magnitude with an arbitrary direction. Thus, no particular care is in practice needed with the base-line between views in the map, other than avoiding the initialization of any pair of optical centers with exactly the same 3D position to prevent division by zero in (5).

Incremental optimization: One of the more interesting research topics raised by the proposed approach is incremental operation. In order to reestimate the optimal solution for the $N+1$ view, we only have to add a new column to the Hessian obtained for the previous N views, coming from just $O(2Kl)$ nonzero new subblocks in the updated Jacobian, being l the average track length and K a bound on its standard deviation. Since the Hessian will remain rather sparse and the previously refined solution is a good starting point, an iterative sparse solver based on the preconditioned conjugate gradient method will expectedly solve the new system of equations faster than a general solver based on complete Cholesky or LDL decomposition. Note also that all the previously computed $\hat{M}_{ij}$ matrices for the rest of views will in any case be reused without ever needing to recompute them. In the bootstrap process of building an incremental solution for large problems, all these facts result in a valuable computational advantage; in real time systems, on the other hand, though we will have to eventually freeze the oldest views and reestimate only the most recent ones (as usual in these kind of systems [7]), the number of free views that can be reestimated in the allocated time will probably be much larger than in standard BA.

Convergence speed: The experiments presented in §6 show that convergence of the GEA algorithm is faster than that of BA implementations. Two or three iterations are usually enough to get a solution very close to the optimum, while full BA require more iterations and approaches more gradually to the solution. The geometric reprojection error is not quadratic and therefore convergence is slow due to cross-sectional local minima [13]. In contrast, the proposed epipolar cost, without structure variables, has a much wider, regularized convergence basin suitable to Newton minimization and tolerant to imprecise initialization.

Limitations: The proposed approach has also some drawbacks. For example, line features do not impose constraints on a view pair, so they cannot be easily included in the epipolar cost. Also, exact treatment of the uncertainty ellipsoids in the algebraic cost (as in [19]) may not be compatible with the computational advantages of the reduced cost (anyway, under the simplification of varying isotropic uncertainty we could still assign an appropriate scalar uncertainty value to each pair of matching points, in order to convert eq. (6) in an ordinary weighted least squares problem). Finally, there is also a critical situation for the epipolar cost, which consists of "isolated chains" of views with collinear centers without links to other views in the "core" of the view

[11] A slight bias of the algebraic cost to "push" the epipoles towards the image center has been reported in [30].

graph. This would add undesired degrees of freedom in the cost function, preventing correct estimation of the scale in their relative baselines. This problem should be detected and fixed postponing the estimation of those views to a last stage, when structure is already available.

8. Conclusion

We have proposed a global epipolar cost function to refine incremental SfM solutions, as a more efficient alternative to the sum of individual reprojection errors used in classical SBA approaches. With better convergence properties, due to early elimination of 3D points in the optimization, this cost function gets solutions remarkably close to the BA optimum in terms of geometric error, safely keeping the incremental reconstruction from eventual divergence at any step. The final solution needs only an ultimate refinement at the very end, with two or three additional iterations of the traditional joint estimation of all the views and 3D points available in the classical SBA framework. Our preliminary results, though still obtained with an unoptimized implementation, show that the approach is particularly interesting for global refinement of the estimated motion and 3D world in real-time systems. We have also discussed a number of promising improvements and possibilities of optimization which are currently the subject of our research.

Acknowledgements

This work was supported by the Spanish MEC and MICINN, as well as European Commission FEDER funds, under Grants CSD2006-00046 and TIN2009-14475-C04. We also thank the reviewers for their useful comments.

References

[1] S. Agarwal, N. Snavely, S. M. Seitz, and R. Szeliski. Bundle adjustment in the large. In *Proc. of ECCV*, 2010.

[2] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski. Building Rome in a day. In *ICCV*, 2009.

[3] A. Bartoli. A unified framework for quasi-linear bundle adjustment. In *Proc. of ICPR*, 2002.

[4] Y. Chen, T. A. Davis, W. W. Hager, and S. Rajamanickam. Cholmod, supernodal sparse cholesky factorization and update/downdate. *ACM Trans. Math. Softw.*, 35:22:1–22:14, October 2008.

[5] T. A. Davis. *Direct Methods for Sparse Linear Systems*. SIAM, Philadelphia, PA, USA, 2006.

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

[7] C. Engels, H. Stewénius, and D. Nister. Bundle adjustment rules. In *In Photogrammetric Comp. Vision*, 2006.

[8] A. W. Fitzgibbon and A. Zisserman. Automatic camera recovery for closed or open image sequences. In *ECCV*, 1998.

[9] G. Golub and C. V. Loan. *Matrix Computations*. The Johns Hopkins University Press, 3^{rd} edition, 1996.

[10] V. M. Govindu. Lie-algebraic averaging for globally consistent motion estimation. In *Proc. of CVPR*, 2004.

[11] N. Guilbert, A. Bartoli, and A. Heyden. Affine approximation for direct batch recovery of euclidean structure and motion from sparse data. *IJCV*, 69(3):317–333, 2006.

[12] Y. Han. Relations between bundle adjustment and epipolar geometry based approaches, and their applications to efficient SfM. *Real-Time Imaging*, 10(6):389–402, 2004.

[13] R. Hartley and F. Schaffalitzky. Powerfactorization: 3d reconstruction with missing or uncertain data. In *Australia-Japan Advanced Workshop on Computer Vision*, 2003.

[14] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[15] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Proc.* 6^{th} *IEEE and ACM Int. Symp. on Mixed and Augmented Reality (ISMAR)*, 2007.

[16] K. Konolige. Sparse sparse bundle adjustment. In *Proc. of BMVC*, 2010.

[17] B. Liu, M. Yu, D. Maier, and R. Männer. An efficient and accurate method for 3D-point reconstruction from multiple views. *IJCV*, 65(3):175–188, 2005.

[18] M. Lourakis and A. Argyros. Is Levenberg-Marquardt the most efficient optimization algorithm for implementing bundle adjustment? In *Proc. of ICCV*, 2005.

[19] M. Lourakis and A. Argyros. SBA: a software package for generic sparse bundle adjustment. *ACM Trans. on Math. Software (TOMS)*, 36(1):2.1–2.30, 2009.

[20] Q. Luong and O. Faugeras. The fundamental matrix: theory, algorithms, and stability analysis. *IJCV*, 17:43–75, 1995.

[21] Y. Ma, S. Soatto, and J. Kosecka. *An invitation to 3-D vision*. Springer Verlag, 2004.

[22] N. Munksgaard. Solving sparse symmetric sets of linear equations by preconditioned conjugate gradients. *ACM Trans. on Math. Software (TOMS)*, 6(2):206–219, 1980.

[23] D. Nister. Reconstruction from uncalibrated sequences with a hierarchy of trifocal tensors. In *Proc. of ECCV*, 2000.

[24] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3d. In *SIGGRAPH*, 2006.

[25] N. Snavely, S. M. Seitz, and R. Szeliski. Skeletal graphs for efficient structure from motion. *Proc. of CVPR*, 2008.

[26] R. Steffen, J.-M. Frahm, and W. Förstner. Relative bundle adjustment based on trifocal constraints. 2010.

[27] J.-P. Tardif, A. Bartoli, M. Trudeau, N. Guilbert, and S. Roy. Algorithms for batch matrix factorization with application to structure from motion. In *Proc. of CVPR*, 2007.

[28] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment - a modern synthesis. In *ICCV: Proc. of International Workshop on Vision Algorithms*, 2000.

[29] R. Vidal, Y. Ma, S. Hsu, and S. Sastry. Optimal motion estimation from multiview normalized epipolar constraint. In *Proc. of ECCV*, 2001.

[30] Z. Zhang. Determining the epipolar geometry and its uncertainty: A review. *IJCV*, 27:161–195, 1998.

[31] Z. Zhang and Y. Shan. Incremental motion estimation through modified bundle adjustment. In *ICIP*, 2003.