National Research Council Canada
Institute for Information Technology

Conseil national de recherches Canada
Institut de technologie de l'information

NRC · CNRC

*Evolutionary Based Autocalibration from the Fundamental Matrix **

Whitehead, A., Roth, G.
April 2002

* published in Applications of Evolutionary Computing, LNCS Springer Verlag 2279, pp. 292-303. Kinsale, Ireland. April 2002. NRC 45863.

Canada

Evolutionary Based Autocalibration from the Fundamental Matrix

Anthony Whitehead[1,*] and Gerhard Roth[2]

[1]School of Computer Science, Carleton University, Ottawa, Canada
awhitehe@scs.carleton.ca

[2]National Research Council of Canada, Ottawa, Canada K1J 6H3
Gerhard.Roth@nrc.ca

Abstract. We describe a new method of achieving autocalibration that uses a stochastic optimization approach taken from the field of evolutionary computing and we perform a number of experiments on standardized data sets that show the effectiveness of the approach. The basic assumption of this method is that the internal (intrinsic) camera parameters remain constant throughout the image sequence, i.e. they are taken from the same camera without varying the focal length. We show that for the autocalibration of focal length and aspect ratio, the evolutionary method achieves comparable results without the implementation complexity of other methods. Autocalibrating from the fundamental matrix is simply transformed into a global minimization problem utilizing a cost function based on the properties of the fundamental matrix and the essential matrix.

1 Introduction

Advances in the field of projective vision make it possible to compute various quantities from an uncalibrated image sequence: in particular the fundamental matrix between image pairs [1, 2]. Autocalibration has become popular due to these recent advances because of the desire to create 3D reconstructions from a sequence of uncalibrated images without having to rely on a formal calibration process. The standard model for an uncalibrated camera has five unknown intrinsic parameters found in a 3x3 calibration matrix K. These parameters are the focal length, aspect ratio, skew and the center of projection x and y (the principal point). The accurate estimation of these 5 parameters is the fundamental goal of autocalibration.

Autocalibration algorithms can be divided into two basic classes. In class A algorithms, we compute the calibration matrix K from the fundamental matrix (the recovered epipolar geometry) [4, 5, 6, 7, 8] and class B algorithms compute K from a projective reconstruction [9, 10, 11] of the scene. Since the projectively reconstructed frames must all be warped to a consistent relative base, Class B algorithms are computationally difficult in comparison to simply finding the fundamental matrix between image pairs. It is claimed that Class B autocalibration algorithms are superior to Class A algorithms because the Class A algorithms do not enforce the constraint that the

[*] Partially funded by Nortel Networks Scholarship.

S. Cagnoni et al. (Eds.): EvoWorkshops 2002, LNCS 2279, pp. 292–303, 2002.

plane at infinity be the same over the entire image sequence [1]. It is precisely this constraint that makes Class B algorithms computationally difficult and we show that Class A algorithms combined with the use of evolutionary systems are as accurate as their Class B counterparts.

Another concern with Class A algorithms is the existence of extra degenerate motions, these being pure rotations, pure translations, affine viewing and spherical camera motions [1, 12]. However, there exist many practical situations that do not contain these degenerate motions where autocalibration is necessary. For example, there are many photographs and video clips in existence for which there is no knowledge of the camera. In order to reconstruct some of these image sequences, autocalibration is the only means.

Autocalibration has been criticized [13] in the past because many different possible calibrations will always provide a 3D reconstruction with almost perfect Euclidean structure. In essence, the only thing we can really measure, the skews and aspect ratios, are very close to what they should be because of manufacturing accuracy. Because of this, the corresponding reconstruction will always look good i.e. the different right angles look square and the different length-ratios look correct. Commonly, the “look” of a reconstruction is used as a ground truth element, but it is clearly a weak one, and any algorithm using such a comparison as a measure of goodness is highly suspect. Because of the manufacturing accuracies, we attempt to autocalibrate only the focal length and the aspect ratio and make assumptions about the remaining parameters.

The constraining equations for the two autocalibration methods presented in this paper are based on the fundamental matrix, and are non-linear. In what follows, we will show that it is possible to reformulate the process of autocalibration as the minimization of a cost function of the calibration parameters. This type of reformulation has not been achieved for all autocalibration algorithms, specifically the class B algorithms which are thought to be superior. For example, the modulus constraint is a non-linear relationship between the camera calibration parameters and the projective camera matrices that have been used as the basis of a class B autocalibration algorithm [10]. The application of the modulus constraint produces a set of polynomial equations for every pair of images, and a system of polynomial equations for the entire image sequence. The solution of such a polynomial system is very difficult to compute, but one possibility is to find all the permutations of exact solutions in closed form and then to combine the results [8]. This is rather cumbersome, and another way to solve such a polynomial system is to use a continuation method [17]. Unfortunately, continuation methods only work well for a small number of equations, and are not suitable for the large polynomial systems that are generated by long image sequences.

In this paper, we examine two autocalibration algorithms that use fundamental matrices and an evolutionary approach to estimating the parameters; one based on Kruppa's equation [6, 4, 8], and the second based on the idea of finding the calibration matrix which optimally converts a fundamental matrix to an essential matrix [7]. In both cases the problem can be formulated as the minimization of a cost function that we describe in sections 2 and 3. The correct camera calibration is the global minimum of this cost function over the space of possible camera parameters. In the past, the claim has been that such minimization approaches to autocalibration are sensitive to the initial starting point of the gradient descent algorithm, but when computing only one parameter, the starting point is irrelevant because we can solve the associated 1D optimization problem using standard numerical approaches [14]. When there is more than one parameter, such as focal length and aspect ratio, we use a simple stochastic approach [15] from the field of evolutionary computing to overcome this problem. We show experimentally that for this type of cost function this stochastic method reliably finds the global minimum. As well, a number of experiments are performed on image sequences with known camera calibration, some of which have been described in the autocalibration literature and utilize class B algorithms. We show that the stochastic approach achieves results that are as good as the class B algorithms. The next section describes the two autocalibration methods, and the theory behind them. The third section describes the experiments, and the fourth presents the conclusions and future avenues of research to improve accuracy.

2 Autocalibration from the Fundamental Matrix

The goal of autocalibration is to compute the camera calibration matrix K. The standard linear camera calibration matrix (K), used to convert from image coordinates in pixels to world coordinates on the camera-sensing element in millimeters, has the following entries [1, 2]:

$$K = \begin{pmatrix} fk_u & -fk_u \cot(\theta) & u_0 \\ 0 & fk_v / \sin(\theta) & v_0 \\ 0 & 0 & 1 \end{pmatrix} \quad \textbf{(1)}$$

Here f is the focal length in millimeters, and k_u and k_v are the number of pixels per millimeter for the camera. If we let α_υ and α_v be fk_u and fk_v respectively by multiplying the focal length *(f)* in mm and the mm/pixel (k), we have the focal length in pixels. The ratio α_υ / α_v is the aspect ratio and is often (but not always) one because of manufacturing, and θ is the skew angle. The skew angle θ is almost always 90 degrees, again because of manufacturing. This leaves us with four free intrinsic camera parameters α_υ , $\alpha_{v,}$ u_0 and v_0. The calibration matrix K can therefore be rewritten in a much simpler form as:

$$K = \begin{pmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \quad \textbf{(2)}$$

The fundamental matrix F is a 3x3 matrix of rank 2 that defines the epipolar geometry between two images [2]. Given two corresponding points m_1 and m_2 from images I_1 and I_2, the epipolar constraint specifies:

$$m_2 F m_1 = 0 \quad \textbf{(3)}$$

The fundamental matrix can be computed from a set of 2D correspondences between the two images [16]. If we know the epipolar geometry and thus the Fundamental Matrix, it is possible to compute the intrinsic camera parameters.

2.1 Autocalibration via Equal Essential Eigenvalues

The essential matrix can be considered as the calibrated version for the fundamental matrix. Given the camera calibration matrix K and the fundamental matrix F, then the essential matrix E is related by the following equation:

$$E = K^T F K \tag{4}$$

Since F is a 3x3 matrix of rank two with the condition that there are exactly two non-zero eigenvalues, E is also of rank two. E however, has an added constraint that the two non-zero eigenvalues must be equal [2]. It is this constraint that is used to create the autocalibration algorithm [7]. The idea is to find the calibration matrix K that makes the two eigenvalues of E equal, or in the case of estimation, as close as possible. Given two non-zero eigenvalues of E, σ_1 and σ_2 where $\sigma_1 > \sigma_2$, then in the ideal case $(\sigma_1 - \sigma_2)$ should be zero. Consider the difference $(\sigma_1 - \sigma_2) / \sigma_1$, which can be written as:

$$1-(\sigma_2/\sigma_1) \tag{5}$$

If the eigenvalues of E are equal, (5) computes to zero; as they differ, equation (5) approaches one. Clearly, (5) becomes the cost function to be minimized.

As we are dealing with a sequence of N images, we can have at most N-1 adjacent image pairs and therefore we have N-1 different fundamental matrices F_i (i=1..N-1). Based on our assumption that the same camera with invariant intrinsic parameters is used, our goal is to find K by minimizing the cumulative values of (5) for all the fundamental matrices F_i in the sequence. Assume F_i is the fundamental matrix relating image I_K and I_{K+1}. To autocalibrate over the N image sequence, we must find the K that minimizes:

$$\sum_{i-1}^{N-1} \omega_i (1 - \sigma 2 / \sigma 1) \tag{6}$$

Where ϖ_i is a weight factor, between zero and one, which defines the confidence of the computed fundamental matrix F_i. ϖ_i is defined in more detail in the next section.

2.2 Autocalibration via Kruppa's Equations

Another way to perform autocalibration from the fundamental matrix is to use Kruppa's equations [1, 2]. To understand these equations we must first define the absolute conic. In Euclidean space the absolute conic lies on the plane at infinity, and has the equation:

$$x^2 + y^2 + z^2 = 0 \,. \tag{7}$$

The absolute conic contains only complex points that satisfy $x^T x = 0$. If we consider a standard camera projection matrix P = K[R|-Rt]. Where R is the rotational motion of between camera positions and t is the translation component of the camera motion, thus a 3D point x on the absolute conic projects to a 2D point:

$$u = P(x) = KRx. \tag{8}$$

Thus, $x = R^T K^{-1} u$, and since $x^T x = 0$, this implies:

$$u^T K^{-1} R R^T K^{-1} u = u^T K^{-T} K^{-1} u = 0 \,. \qquad \textbf{(9)}$$

This clearly shows that any 2D point u is on the image of the absolute conic if and only if it lies on the conic represented by the matrix $K^{-T}K^{-1}$. From projective geometry, KK^T is the dual absolute conic, and is labeled as C. If we can find C, then we can directly compute the camera parameters K by Cholesky factorization.

Kruppa's equations relate the fundamental matrix to the terms of the dual absolute conic. The first form of these equations required the computation of not just the fundamental matrix, but also of the two camera epipoles, which are known to be unstable [2]. Recently, a new way of relating the fundamental matrix and the dual absolute conic was described which does not require the computation of the camera epipoles [4]. Consider the singular value decomposition of a fundamental matrix F to be UDV^T. We let the column vectors of U and V be u_1, u_2, u_3 and v_1, v_2, v_3 respectively. This gives the new form of Kruppa's equation to be:

$$\frac{v_2^T C v_2}{r^2 u_1^T C u_1} = \frac{-v_2^T 2C v_1}{sru_1^T C u_1} = \frac{v_1^T C v_1}{s^2 u_2^T C u_2} \qquad \textbf{(10)}$$

To autocalibrate we must find the C which makes these three ratios equal, or in the case of estimation, as close to equal as possible. We let $ratio_1$ be equal to:

$$\frac{v_2^T C v_2}{r^2 u_1^T C u_1} - \frac{-v_2^T 2C v_1}{sru_1^T C u_1} \qquad (11)$$

And define $ratio_2$ and $ratio_3$ similarly as the other two possible permutations in the ratios. Autocalibration can then be achieved by finding the C (KK^T) that minimizes the sum of the ratios squared. Given the same image sequence that produced equation 6, the Kruppa ratios over n images minimizes:

$$\sum_{i-1}^{N-1} \omega_i (ratio_1^2 + ratio_2^2 + ratio_3^2) \qquad \textbf{(12)}$$

Again, ϖ_i is a weight factor, between zero and one, which defines the confidence of the computed fundamental matrix F_i.

2.3 Evolutionary Idea

Since the two autocalibration methods based on the fundamental matrix have an associated cost function we can use a gradient descent algorithm to find the solution. The caveat here is that there are often many local minima in the cost function, so the solution that is found depends on the starting point. However, we note that the calibration parameters can all be bounded; i.e. the center of projection rarely varies from the image center, the aspect ratio is generally 1 and the skew is 90 degrees. Thus we are attempting to find the global minimum for a set of real-valued, bounded optimization parameters. This problem has been dealt with in the field of evolutionary computing. We use an approach called Dynamic Hill Climbing (DHC), that combines genetic algorithms, hill climbing and conjugate gradient methods to be an optimization algorithm that is very successful in solving such real valued optimization problems [15].

The idea is to repeatedly perform gradient descent in the search space and to restart the gradient descent in an area of the search space that is as far removed as possible from previous solutions. We call such a method Statistically Distributed Random Starting (SDRS), and in this way we cover the search space as effectively as possible as seen in Fig 1.

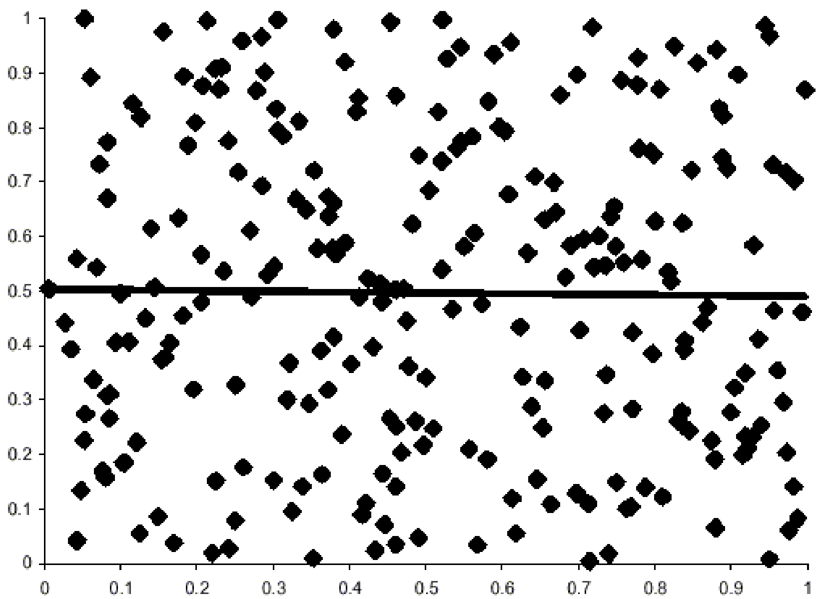

Fig. 1. : Scatter plot of 2D search space generated by SDRS. 250 points with atrend line indicating an even disbursement of start points.

The pseudo-code for SDRS:

```
SDRS()
For each optimization parameter in the search space
    Find the largest region that has not had a start point
    Compute a random point in this region
    Set this point to the start point for this dimension
Endfor
Return N-dimensional startPoint
```

SRDS allows for the most complete coverage of the search space with a user specified number of runs. This allows the DHC algorithms to successfully find the global minimum throughout the search space.

The pseudo-code for estimating K:

```
ESTIMATE_K()
For n times
    StartPoint = SRDS()
    Perform the DHC gradient descent from StartPoint.
    IF Cost function (Equal Eigenvalues OR Kruppa) is minimal
        Save this K.
    ELSE
        Discard this K
    Endfor
Return K
```

The algorithm ESTIMATE_K returns the calibration parameters in the matrix K that produced the minimum value from the cost function. Evaluating the cost function for the two different autocalibration methods is very efficient. A single gradient descent of the cost function uses the Powell optimization algorithm, which is in turn based on repeated applications of the one dimensional Brent method [14]. The equal eigenvalues approach requires only the computation of the eigenvalues of a three by three matrix, and for the Kruppa approach the computation of three ratios. In both cases the weights ω_i are set in proportion to the number of matching 2D feature points that support a given fundamental matrix. The larger the number of points that support the epipolar geometry characterized by F, the more confidence we have in that fundamental matrix, and therefore the greater the weight. We next show experimentally that the global minimum is found very reliably by this approach.

3 Experimental Results

For many autocalibration algorithms the evaluation of performance consists of a simple visual inspection of the resulting 3D reconstruction. This is not adequate because it has been shown that the quality of the final reconstruction is visually acceptable for a wide variety of calibration parameters [13]. In order to compare the capabilities of the evolutionary method, we performed a variety of experiments that compared against the results from the literature that used “look” as a goodness criteria. This allowed us to compare the evolutionary method against other algorithms, and specifically we show that it has comparable results to the more complicated Class B algorithms. A secondary measure for experimentation is comparison against ground truth, i.e. the intrinsic parameters are already known a-priori. Finally, we take several sequences taken from the same uncalibrated camera and show that the evolutionary computing based algorithm is consistent and repeatable.

The first set of experiments described in Table 1 show how the autocalibration process works when we are calibrating only the focal length. Table 1 shows the results for a number of different test sequences that have been processed in previous autocalibration papers [6, 8, 10, 19]. In particular, the castle sequence is used as a test case for comparison of the class B approach that requires a projective reconstruction [10]. We see that our autocalibration results are comparable to those of other algorithms.

Table 1: Results of autocalibration for focal length vs other algorithms. Focal length is in pixels. Correspondences are computed automatically.

Name	# of Images	Stated Focal	Computed focal len (Eq.Eigen)	% error vs. Stated	Computed focal len (Kruppa)	% error vs. Stated
Castle	27	1100	1156.50	5	1197.7	8
Valbone	9	682	605.5	11	685.71	0.5
Nekt	6	700	798.58	14	872.44	24.6
etluueshiba	5	837	857.25	2.4	1233.85	47.4

It is important to note that the stated focal lengths are those computed in the literature, and the assessment of goodness was how the reconstruction looked. In the table we compare how close our autocalibration results are to the previously published results, which we assume to be reasonably correct but cannot confirm. In the last example shown in the table [19], the error with the Kruppa autocalibration is quite large, possibly because the motion is close to being a pure translation which is known to be a degeneracy motion for the Kruppa algorithm [1,12]. It is also a good indicator of how the Equal Eigenvalues method performs well against these degenerate motions.

Finally, because the ground truth is not really known, and the methods for computing F in the literature are not available, it is possible that the stated focal lengths are incorrect.

In the set of experiments outlined in Table 2, the 2D feature points were selected by hand as part of a photogrammetric model building process. From these manually selected correspondences we compute the fundamental matrix between all image pairs in the sequence. In this experiment we know the intrinsic parameters of the camera a-priori from the projects of the photogrammetric package [20]. We therefore assume that all the intrinsic parameters are set a-priori, except for the focal length which we autocalibrate.

Table 2: Results of autocalibration for focal length for photogrammetric sequences. Focal length is in mm., and reprojection error is in pixels. Correspondences selected by hand.

Name	# of Images	True focal	Eigen focal	% error	Kruppa focal	% error	Correct reproj.	Eigen reproj.	Kruppa Reproj.
Curve	4	6.97	4.71	32.4	7.49	1.13	7	2.23	1.44
Cylinder	3	28	26.35	5.9	31.70	13.21	0.96	2.07	2.60
Plant	6	24.20	22.55	6.8	24.39	0.78	0.80	1.49	1.04
Statue	7	5.11	3.67	28.2	5.29	3.5	3.93	9.61	1.95

Table 2 shows the autocalibrated focal length in millimeters versus the true focal length, along with the percentage error for both autocalibration methods. Since we have the associated 3D reconstructions for the corresponding 2D features we can compute more sophisticated performance measures. For a given autocalibrated focal length we compute the reprojection error for all the corresponding feature points. The reprojection errors are the pixel differences between the projection of the 3D feature points into 2D and the original corresponding 2D features. We compute the median of the reprojection errors using the correct focal length, the focal length found by the eigenvalue method, and the focal length found by Kruppa's method. The median of the reprojection errors is a good indicator of the quality of the reconstruction for a given focal length. We see that the median reprojection error increases for the autocalibrated focal lengths, but only slightly. This implies that the error in the autocalibrated focal lengths would not have a significant impact in terms of reconstruction quality and this independently verifies the claims of Bougnoux [13].

In the next experiment we attempt to autocalibrate both aspect ratio and focal length using the two methods. We are again using as input a series of photogrammetric projects for which we know the 2D feature correspondences as well as the ground truth of the intrinsic camera parameters.

Table 3: Results of autocalibration for focal length and aspect ratio for photogram-metric sequences. The equal eigenvalue method is used and focal length is in mm.

Name	True aspect	Eigen Aspect	Variance	% error	True Focal	Eigen focal	Variance	% error
Curve	1.0	1.08	0.003	8	6.97	3.46	0.062	50
Cylinder	1.0	0.98	0.002	2	28	26.72	0.52	4.5
Plant	1.0	0.98	0.012	2	24.2	22.96	0.39	5.1
Dam	0.81	0.972	0.0001	20	30.75	38.52	0.089	9.8

While the results as shown in Tables 3 and 4 are reasonable, the errors when autocalibrating two camera parameters are sometimes higher than autocalibrating just one parameter. The error again compounds when we attempt to autocalibrate all parameters. In particular, the percentage error in focal length increases slightly. One possible explanation is that the gradient descent algorithm is stuck in a local minima, to verify this the results shown in these two tables were computed by averaging over one hundred separate runs of the optimization algorithm. The variance as shown in the table for the autocalibrated aspect ratio and focal length is very small over these runs and indicates that it is highly likely that the stochastic optimization algorithm is finding the correct global minimum.

Table 4: Results of autocalibration for focal length and aspect ratio for photogrammetric sequences. The Kruppa autocalibration method is used.

Name	True aspect	Kruppa Aspect	Variance	% error	True Focal (mm)	Kruppa focal (mm)	Variance	% error
Curve	1.0	0.997	0.011	1.3	6.97	7.56	0.21	8.4
Cylinder	1.0	1.03	0.0001	3	28	32.91	0.0001	17.5
Plant	1.0	0.92	0.003	8	24.2	26.33	0.12	8.8
Dam	0.81	0.997	0.0001	19.75	30.75	38.43	0.0001	24.9

The final experiment, as shown in Table 5, has as input three image sequences that were taken with the same camera with invariant intrinsic parameters.

Table 5: Results for autocalibration of focal length for three sequences

Name	# of Images	Eigen focal	Kruppa Focal
Chapel	12	27.82	31.31
Climber	13	27.91	33.88
Workshop	8	26.19	38.09

Test cases chapel and workshop are almost pure translation while the climber sequence has a motion with significant translation and rotation. We autocalibrate only the focal lengths, which should be equal for all three sequences. The variance of the computed focal length for the eigenvalue method is 0.96 mm and for Kruppa approach is 3.42mm. It is not surprising that the autocalibration results differ, since certain motions are degenerate with regards to the Kruppa based autocalibration [1]. What these results clearly show is that for a given camera, and substantially different sequences, the evolutionary algorithms (especially the equal eigenvalues method) are convergent.

In summary, the experiments show that the evolutionary approach is as good as any complicated Class B algorithm, e.g. the castle sequence in Table 1. Computationally the fundamental matrix based approaches are very efficient since a single evaluation of the cost functions does not take long. The time taken for autocalibration is in the order of seconds for all the image sequences on a 400 MHz Pentium II processor. It also becomes clear that the equal eigenvalues method is superior to the Kruppas method for degeneracy cases. There are cases, however, where the Kruppas method is clearly outperforming the equal eigenvectors method. Further investigation is necessary to determine whether or not a heuristic can be developed to choose one algorithm over the other by pre-determining the camera motion.

4 Conclusions

In theory the autocalibration methods that use fundamental matrices should not perform as well as those that use the camera projection matrices of a projective reconstruction [1, 12, 2]. However, we show that for non-degenerate motions both methods perform equally well when we are calibrating only the focal length, or the focal length and aspect ratio. Similarly in [11], the principle point was not computed accurately using the class B algorithm and was also subsequently assumed.

The equal eigenvalues approach is very simple and works just as well as any Class B method we compared against. While it is theoretically equivalent to the Kruppa approach, it performs better numerically in situations where we are close to a degenerate motion, such as pure translation. The usual class B approach to autocalibration requires the solution of a set of polynomial equations but this is not computationally feasible for long image sequences. With our evolutionary computing based approach we can process long image sequences, which is an advantage to these algorithms. The argument against the optimization-based methods has been that they are sensitive to the starting point of the optimization process [3, 5]. We have shown that the SDRS method helps to find the global minimum of the cost function reliably. In our experiments we have shown that the error in the autocalibration of the focal length is usually in the range of 5% to 15%. This is adequate for applications in which the final results are used for visualization purposes, such as model building but clearly not for applications that require exact depth information.

What may not be an obvious next step is to move forward and decrease the error is to utilize the two (and more as they become available) autocalibration routines (equal eigenvalues and Krupps equations) in yet another evolutionary step. In essence this means that we want to minimize the difference between two calibration matrices K_{KRUPPA} and K_{EIGEN}. This can be measured in a variety of ways, but clearly the

Frobineus norm measure does this exact difference assessment for us. As well, results may become more stable by performing the SDRS algorithm in a windowed manner to ensure better coverage of the search space.

Acknowledgements

Thanks to Photomodeler corporation [20] for providing the image sequences for the photogrammetric projects.

References

[1] O. Faugeras and Q.-T. Luong, The Geometry of Multiple Images. The MIT Press, 2001.
[2] R. Hartley and A. Zisserman, Multiple view geometry in computer vision. Cambridge University Press, 2000..[3] A. Fusiello, Uncalibrated Euclidean reconstruction: a review," Image and Vision Computing, vol. 18, pp. 555-563, 2000.
[3] A. Whitehead and G. Roth, The Projective Vision Toolkit, in Proceedings Modelling and Simulation, (Pittsburgh, Pennsylvania), May 2000.
[4] R. Hartley, “Kruppa's equations deri ved from the fundamental matrix," IEEE Trans. On Pattern Analysis and Machine Intelligence, vol. 19, pp. 133-135, February 1997.
[5] Q.-T. Luong and O.D.Faugeras, “Self-calibration of a moving camera from point correspondences and fundamental matrices," International Journal of Computer Vision, vol. 22, no. 3, pp. 261-289, 1997.
[6] L. Lourakis and R. Deriche, “Camera self-calibration using the svd of the fundamental matrix," Tech. Rep. 3748, INRIA, Aug. 1999.
[7] P. Mendonca and R. Cipolla, “A simple technique for self-calibration," in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, (Fort Collins, Colorado), pp. 112-116, June 1999.
[8] C. Zeller and O. Faugeras, “Camera self-calibration from video sequences: the kruppa equations revisited,” Tech. Rep. 2793, INRIA, Feb. 1996.
[9] M. Pollefeys, R. Koch, and L. V. Gool, “Self calibration and metric reconstruction in spite of varying and unknown internal camera parameters," in International Conference on Computer Vision, pp.90-96, 1998.
[10] M. Pollefeys, Self-calibration and metric 3d reconstruction from uncalibrated image sequences. PhD thesis, Catholic University Leuven, 1999.
[11] M. Pollefeys, R. Koch, and L. V. Gool, “Self-calibration and metric reconstruction in spite of varying and unknown intrinsic camera parameters," International Journal of Computer Vision, vol. 32, no. 1, pp. 7-25, 1999.
[12] P. Sturm, A case against kruppa's equations for camera self-calibration," IEEE Trans. On Pattern Analysis and Machine Intelligence, vol. 22, pp. 1199-1204, Oct. 2000.
[13] S. Bougnoux, “From projecti ve to Euclidean space under any practical situation, a criticism of self-calibration," in Proc. 6th Int. Conf. on Computer Vision, (Bombay, India), pp. 790-796, 1998.
[14] W. H. Press and B. P. Flannery, Numerical recipes in C. Cambridge university press, 1988.
[15] M. Maza and D. Yuret, “Dynamic hill climbing," AI Expert, pp. 26{31, 1994.
[16] R. Hartley, “In defense of the 8 point algorithm," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 6, 1997.
[17] A. Morgan, Solving polynomial systems using continuation for science and engineering. Prentice Hall, Englewoord Clifis, 1987

[18] G. Roth and A. Whitehead ”Using projective vision to find camera positions in an image sequence," in Vision Interface 2000, (Montreal, Canada), pp. 87-94, May 2000.
[19] T. Ueshiba and F. Tomita, “A factorization method for projective and Euclidea reconstruction," in ECCV'98, 5th European Conference on Computer Vision, (Freiburg, Germany), pp. 290-310, Springer Verlag, June 1998.
[20] Photomodeler by EOS Systems Inc. http://www.photomodeler.com.