EURASIP Journal on Applied Signal Processing 2004:8, 1113–1124

Estimating Intrinsic Camera Parameters from the Fundamental Matrix Using an Evolutionary Approach

Anthony Whitehead
School of Computer Science, Carleton University, Ottawa, ON, Canada K1S 5B6
Email: awhitehe@scs.carleton.ca

Gerhard Roth
National Research Council of Canada, Ottawa, ON, Canada K1A 0R6
Email: gerhard.roth@nrc.ca

Received 30 June 2002; Revised 21 October 2003; Recommended for Publication by Stefano Cagnoni

Calibration is the process of computing the intrinsic (internal) camera parameters from a series of images. Normally calibration is done by placing predefined targets in the scene or by having special camera motions, such as rotations. If these two restrictions do not hold, then this calibration process is called autocalibration because it is done automatically, without user intervention. Using autocalibration, it is possible to create 3D reconstructions from a sequence of uncalibrated images without having to rely on a formal camera calibration process. The fundamental matrix describes the epipolar geometry between a pair of images, and it can be calculated directly from 2D image correspondences. We show that autocalibration from a set of fundamental matrices can simply be transformed into a global minimization problem utilizing a cost function. We use a stochastic optimization approach taken from the field of evolutionary computing to solve this problem. A number of experiments are performed on published and 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, that is, the images are taken from the same camera without varying such quantities as the focal length. We show that for the autocalibration of the focal length and aspect ratio, the evolutionary method achieves results comparable to published methods but is simpler to implement and is efficient enough to handle larger image sequences.

Keywords and phrases: autocalibration, dynamic hill climbing, fundamental matrix, evolutionary computing, epipolar geometry, 3D reconstruction.

1. INTRODUCTION

Calibration is the process of computing internal physical quantities of a camera's geometry. Parameters such as focal length, center of projection, and CCD sensor array dimensions are required in order to get 3D information from a series of images. Autocalibration has become popular recently 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 calibration model for a pinhole camera has five unknown intrinsic parameters defined in a 3×3 calibration matrix (K). These parameters are the focal length, aspect ratio, sensor skew, and the center of projection x and y (the principal point). The accurate estimation of these 5 parameters directly from an image sequence without having a formal calibration process is the goal of autocalibration.

Autocalibration works by computing aforementioned quantities directly from 2D image correspondences, and then using invariants of these quantities to find the camera calibration. The fundamental matrix and the full projective reconstruction are two quantities that can be computed from a set of 2D image correspondences, and they are the basis of most autocalibration algorithms. As such autocalibration algorithms can be divided into three classes that we will refer to as classes A, B, and C. In class A algorithms, we compute the calibration matrix K from the fundamental matrix (the recovered epipolar geometry) [1, 2, 3, 4, 5]. In class B algorithms (K) is computed from a projective reconstruction [6, 7, 8] of the scene. Class C algorithms auto calibrate from homographies and planar features within an image sequence [9, 10].

While class C algorithms can compute intrinsic camera parameters from a set if interimage homographies [11], we

loosely consider them autocalibration routines. Because a homography is a planar transformation, class C algorithms require the use of planar targets [12] or the automatic detection and correspondence of planar regions within an image sequence. While it has been shown that planar regions may be robustly detected in images [13], it is highly probable that an image sequence will exist where there are no planar objects, or the existing planar objects are not suitable for robust detection. The aforementioned requirements must be known a priori for computing the calibration parameters, and therefore class C algorithms are not generalized, rather they rely on specific features that may not be present. Due to these facts, it is questionable whether or not a class C algorithm is truly an autocalibration routine in the sense that it requires a target (therefore not autocalibration), or is presupposed by the planar region detection/correspondence problem (therefore not generalized). Because of these problems, class C algorithms are not considered in this work.

In this work we compare against class B algorithms which are thought to be numerically superior to other calibration methods. 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 often claimed that class B autocalibration algorithms are superior to class A and class C algorithms because those algorithms do not enforce the constraint that the plane at infinity (an invariant between projective and Euclidean space) should be the same over the entire image sequence [14]. It is precisely this constraint that makes class B algorithms computationally difficult. In this work, we provide evidence that class A algorithms combined with the use of evolutionary systems produce as accurate an autocalibration 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 [14, 15]. However, there exist many practical situations that do not contain these degenerate motions. Also, in many cases autocalibration is the only option, and even a less accurate autocalibration result is better than no calibration at all. For example, there are many photographs and video clips in existence for which there is no knowledge of the camera. In order to reconstruct the 3D world from those image sequences, autocalibration is the only option.

Autocalibration has been criticized in the past [16] because many different calibrations will provide a 3D reconstruction with reasonable Euclidean structure. In other words, the corresponding reconstruction will usually look good because the different right angles look square and the different length ratios look correct. However, this depends considerably on the image sequence and the camera used to acquire that sequence. All that we can conclude from this fact is that using the "look" of a reconstruction to evaluate the autocalibration results is unreasonable. It is necessary to have the ground truth camera calibration to do a proper performance evaluation. In this paper we evaluate the proposed autocalibration algorithms on image sequences for which the ground truth camera calibration is known a priori as well as comparing against results of class B algorithms.

The constraining equations for the two autocalibration methods presented in this work are nonlinear and based on the fundamental matrix. In what follows, we will show in depth that it is possible to reformulate the process of autocalibration into the minimization of a cost function of the calibration parameters [17, 18]. While this type of reformulation has been achieved for class A algorithms and is clearly evident in class C algorithms, this is not the case for class B algorithms. For example, in [7] the basis of the class B autocalibration algorithm is the modulus constraint. The modulus constraint is a nonlinear relationship between the camera calibration parameters and the projective camera matrices that makes autocalibration possible [6]. The application of the modulus constraint produces a set of X polynomial equations for every pair of images, and a system of polynomial equations for the entire image sequence. Given an M image sequence, we have X^{M-1} equations in the system. The solution of such a polynomial system is very difficult to compute. One possibility is to find all the permutations of exact solutions in closed form and then to combine the results [5]. This is rather cumbersome. Another way to solve such a polynomial system is to use a continuation method [19]. Unfortunately, continuation methods only work well for a small number of equations, and are not suitable for the large polynomial systems generated by long image sequences. By contrast, the methods presented in this work are computationally efficient (with a known upper bound on the number of times the cost function will be executed) even for large image sequences. Furthermore, the accuracy of these algorithms improves as the image sequence lengths increase.

In this work, we examine two class A autocalibration algorithms based on the fundamental matrices, one based on Kruppa's equation [1, 3, 5], and the second based on the idea of finding the calibration matrix which optimally converts a fundamental matrix to an essential matrix [4]. In both cases the problem can be formulated as the minimization of a cost function of the calibration parameters, which will be described in detail in Sections 3 and 4. The correct camera calibration is the global minimum of this cost function over the space of possible camera parameters. In the past, claims have been made that such minimization approaches to autocalibration are sensitive to the initial starting point of the gradient descent algorithm [2, 20]. However, when computing only one parameter, the starting point is irrelevant because we can accurately solve the associated 1D optimization problem using standard numerical approaches [21]. When there is more than one parameter, such as focal length and aspect ratio, we use a simple stochastic approach [22] from the field of evolutionary computing to overcome this problem. We show experimentally that for this type of cost function, the stochastic method reliably finds the global minimum. As well, a number of experiments are performed on image sequences with known camera calibration. We compare the results of our method against class B results on some of the same image sequences, and provide evidence that shows that the stochastic approach achieves results that are comparable.

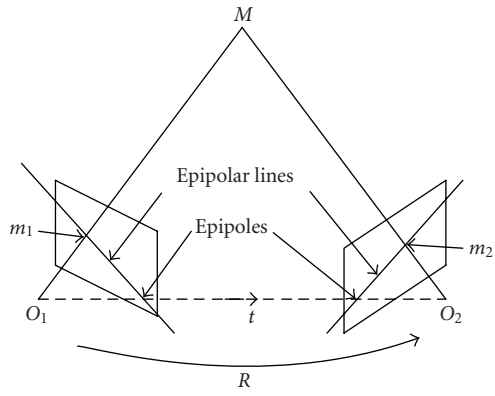

Figure 1: Epipolar geometry of two cameras (O and O') in an arbitrary position, view arbitrary scenes.

This paper continues by providing a brief description of epipolar geometry, followed by a summary of two class A algorithms for computing the fundamental matrix in Section 3. In Section 4 we continue by outlining our method that combines class A algorithms with evolutionary systems. In Section 5, we outline our experimental results and follow up with conclusions.

2. BACKGROUND

To explain the basic ideas behind the projective paradigm, we must first define some notation. We work in homogeneous coordinates, which are defined as an augmented vector created by adding one as the last element. A projection of a point (in the homogeneous Euclidean coordinate system) $M = [X, Y, Z, 1]^T$ to a point m on image plane can be described using the following standard equation:

$$sm = \mathbf{P}M. \tag{1}$$

Here s is an arbitrary scalar, $\mathbf{P}$ is a 3×4 projection matrix, and $m = [x, y, 1]^T$, a 2D homogeneous point on the image plane.

Knowing the camera calibration simply enables us to easily move from a projective space into Euclidean space. This requirement spawned much research into autocalibration techniques.

2.1. The fundamental matrix

The fundamental matrix F is a 3×3 matrix of rank two that defines the epipolar geometry between two images from uncalibrated cameras [23] and characterizes the position of the two cameras independent of the scene structure. Consider a point in 3D space, $M = [X, Y, Z, 1]^T$, and its projected image in two different camera locations (Figure 1), $m_1 = [x_1, y_1, 1]^T$ and $m_2 = [x_2, y_2, 1]^T$. Then the epipolar constraint is

$$m_1{}^T \mathbf{F} m_2 = 0. \tag{2}$$

The fundamental matrix can be computed from a set of corresponding 2D points between the two images. This process is considered to be overly sensitive to noise when compared to iterative methods [24, 25], but in fact a simple preprocessing data normalization step improves the accuracy and produces good results [26].

2.2. Intrinsic camera parameters

If a camera is calibrated, then the calibration matrix ($\mathbf{K}$), containing the internal parameters of this camera (focal length, pixel dimensions, etc.), is known. Using this calibration matrix ($\mathbf{K}$), we can generate the actual 2D image coordinates on the camera-sensing element.

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 [14, 23]:

$$K = \begin{pmatrix} fk_u & -fk_u \cot(\theta) & u_0 \\ 0 & \dfrac{fk_v}{\sin(\theta)} & v_0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{3}$$

Here f is the focal length in millimeters, k_u and k_v are the number of pixels per millimeter (width and height, respectively), and u_0, v_0 are the center of projection. If we let α_u and α_v be fk_u and fk_v respectively by multiplying the focal length (f) in mm by k, given in mm/pixel, this allows us to work in pixel units. The ratio α_u/α_v is now the aspect ratio and is often (but not always) one. The skew angle θ is almost always 90 degrees because modern camera-sensing elements are manufactured accurately. Making these basic assumptions leaves us with four free intrinsic camera parameters α_u, α_v, u_0, and v_0. The calibration matrix K can therefore be rewritten in a much simpler form as

$$K = \begin{pmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{4}$$

where the focal lengths (α_u and α_v), and principal point (u_0, v_0) are all quantified in pixels.

It has been shown [16] that autocalibrating the center of projection u_0, v_0 is not practically useful. For this reason, in this work, we attempt to autocalibrate only the focal length and the aspect ratio and assume that the center of projection is set to be the center of the image. However, results are encouraging when autocalibrating all 4 (focal length, aspect ratio, principal point u and v) intrinsic camera parameters.

2.3. The essential matrix

The essential matrix can be considered the calibrated form of the fundamental matrix. It also encodes the epipolar geometry between two camera views and the epipolar constraint still holds given two points p_1 and p_2 in the camera coordinate system:

$$p_1{}^T \mathbf{E} p_2 = 0, \tag{5}$$

where

$$\mathbf{E} = [\mathbf{t}] \times \mathbf{R}, \tag{6}$$

where $\mathbf{t}$ is the translational motion (vector) between the 3D camera positions, and $\mathbf{R}$ is the rotational motion (matrix) (see Figure 1). The essential matrix can also be computed from a set of camera coordinate correspondences between two different calibrated cameras [27].

A side effect of computing the essential matrix is the Euclidean 3D location of the corresponding points and the camera positions. This is also true for the fundamental matrix, but these coordinates are found in a projective space. The camera position is also found when computing $\mathbf{F}$, but again, only in a projective space.

2.4. The absolute conic

An important concept for autocalibration is the invariant nature of the image absolute conic on multiple image frames. Because it is invariant under Euclidean transformations, its relative position in multiple camera frames remains constant for constant intrinsic camera parameters. The absolute conic has the equation

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

The absolute conic can be seen as a calibration object that occurs in all views of a scene, and once located can be used to compute the intrinsic camera parameters [6].

3. AUTOCALIBRATION FROM THE FUNDAMENTAL MATRIX

Our first class A algorithm relies on the fact that the fundamental matrix can be decomposed into terms of the essential matrix and the camera calibration matrices. Our second algorithm relies on the existence of the projection of the absolute conic within an image pair.

3.1. Autocalibration via equal eigenvalues

Single image pairs

The essential matrix can be considered as the calibrated version of 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:

$$\mathbf{E} = \mathbf{K}^T\mathbf{F}\mathbf{K}. \tag{8}$$

Since F is a 3×3 matrix of rank two with the condition that there are exactly two nonzero eigenvalues, E is also of rank two. The essential matrix (E) however has an added constraint that the two nonzero eigenvalues must be equal [23]. It is this constraint that is used to create the autocalibration algorithm [4]. The goal is to find the calibration matrix K that makes the two eigenvalues of E equal, or as close to equal as possible. Given two nonzero eigenvalues of E, σ_1 and σ_2 where $\sigma_1 > \sigma_2$, in the ideal situation $(\sigma_1 - \sigma_2)$ should be zero. Consider the difference $(\sigma_1 - \sigma_2)/\sigma_1$, which can be rewritten as

$$1 - \left(\frac{\sigma_2}{\sigma_1}\right). \tag{9}$$

If the eigenvalues of E are equal, (9) computes to zero; as they differ, (9) approaches one. Clearly, (9) becomes the cost function to be minimized.

Multiple image pairs

Since we are dealing with a sequence of M images, we can have at most $M - 1$ adjacent image pairs. Since a fundamental matrix is computed between each adjacent image pair, we therefore have $M - 1$ different fundamental matrices F_i $(i = 1, \ldots, M - 1)$. Based on our assumption that the intrinsic parameters of the camera do not vary, our goal is to find K by minimizing the cumulative values of (9) for all the fundamental matrices (F_i) in the sequence. Assume F_i is the fundamental matrix relating images I_K and I_{K+1}. To autocalibrate over the M image sequence, we must find the K that minimizes

$$\sum_{i=1}^{M-1} \omega i \left(1 - \frac{\sigma_2}{\sigma_1}\right), \tag{10}$$

where ωi is a weighting factor, between zero and one, which defines the confidence we have in the computed fundamental matrix F_i. 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 2D points that support the epipolar geometry characterized by F, the more confidence we have in that fundamental matrix, and therefore the smaller the weight (remember we are minimizing). Each weight ω_i is normalized to a range from zero to one.

3.2. Autocalibration via Kruppa's equations

In a similar manner, we can convert Krupp's equations into a cost function that can be used in either single or multiple image pairs.

Single image pairs

Another way to perform autocalibration from the fundamental matrix is to use Kruppa's equations [14, 23]. 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

$$x^2 + y^2 + z^2 = 0. \tag{11}$$

The absolute conic contains only complex points that satisfy the equation $M^TM = 0$. If we consider a standard camera projection matrix

$$P = K[R| - Rt], \tag{12}$$

where R is the rotational component of the motion between camera positions and $-Rt$ is the translational component of

the camera motion, then a 3D point x on the absolute conic projects to a 2D point:

$$m = P(M) = KRM, \tag{13}$$

where

$$M = R^T K^{-1} m; \tag{14}$$

and since $M^T M = 0$, this implies

$$m^T K^{-1} R R^T K^{-1} m = m^T K^{-T} K^{-1} m = 0. \tag{15}$$

This clearly shows that any 2D point m is on the image of the absolute conic if and only if it lies on the conic represented by the matrix

$$K^{-T} K^{-1}. \tag{16}$$

From projective geometry the dual absolute conic for (16) is given by

$$KK^T \tag{17}$$

and is often labeled as C. If we can find C, then we can directly compute the camera parameters K by Cholesky factorization [28].

Kruppa's equations relate the fundamental matrix to the terms of the dual absolute conic. The first form of these equations required the computation not just of the fundamental matrix, but also of the two camera epipoles, which are known to be unstable [23]. 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 [1]. 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 as

$$\frac{v_2^T C v_2}{r^2 u_1^T C u_1} = \frac{-v_2^T 2C v_1}{s r u_1^T C u_1} = \frac{v_1^T C v_1}{s^2 u_2^T C u_2}. \tag{18}$$

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 factor_1 be equal to

$$\frac{v_2^T C v_2}{r^2 u_1^T C u_1} - \frac{-v_2^T 2C v_1}{s r u_1^T C u_1}, \tag{19}$$

and we define factor_2 and factor_3 similarly as the other two possible permutations of the system of ratios. Autocalibration can then be achieved by finding the C (KK^T) that minimizes the sum of the factors squared.

Multiple image pairs

Given the same $M - 1$ fundamental matrices defined in the previous section, then autocalibration with the Kruppa method over M images requires the minimization of

$$\sum_{i=1}^{N-1} \omega i (\text{factor}_1^2 + \text{factor}_2^2 + \text{factor}_3^2). \tag{20}$$

Again, ωi is a weight factor, between zero and one, which is the confidence in the computed fundamental matrix F_i as described in the previous section.

4. THE EVOLUTIONARY APPROACH

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; that is, the center of projection rarely varies from the image center, the aspect ratio is generally one, and the skew is almost always 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.

Experimentally, local gradient descent algorithms that start from different points in the search space do not converge to the same global minimum. We can therefore comfortably conclude that there must exist a number of local minima. Because of this, we need an evolutionary approach that can handle such a situation because any local search algorithm will converge prematurely at a local minimum. We use an evolutionary approach that can find the global minimum, which is the best of the set of local minima.

There are many possible evolutionary approaches, but they are not all equally applicable to every problem. We use the ideas around genetic algorithms (GAs) [29]. The idea behind GAs is to simulate evolution by defining each solution as a chromosome, and then defining the appropriate crossover and mutation operators. While GAs are a very powerful framework, they must be adapted and tuned specifically for each application. In our application of function minimization, the process of simulated annealing has also been successful [17]. The idea behind simulated annealing is to perform function optimization by simulating the process of annealing crystals, essentially by slowly lowering the temperature. The issue we face is, which evolutionary approach is best? We define this problem to mean the simplest and most effective algorithm that arrives at the correct answer.

As the camera calibration problem is being recast as a parameter optimization problem for a set of real-valued, bounded optimization parameters, we use the dynamic hill climbing technique that combines the strengths of GAs and hill climbing techniques that was specifically designed for this type of problem. Dynamic hill climbing (DHC) can be considered a hybrid evolutionary algorithm because the algorithm makes use of concepts such as fitness, population expansion, and mutation, but utilizes a hill climbing technique for determining local extrema. Also, by using a mutating coordinate frame combined with local extrema exploitation, DHC has been empirically shown to outperform classical GAs, simulated annealing, and typical hill climbers when optimizing parameters of the De Jong [30] test suite [22]. DHC optimization results on the De Jong test suite were independently confirmed in [31] and subsequently used in

range image registration. The compared methods included genetic algorithms, simulated annealing and the DHC algorithm. Experimental results showed that the DHC algorithm was the most successful evolutionary approach for this type of bounded, real-valued function optimization. For the above reasons, we choose DHC and we describe the dynamic hill-climbing algorithm in detail next.

4.1. Dynamic hill climbing

The workhorse behind the DHC algorithm is simple yet very efficient hill climbing algorithm; and the use of population expansion via mutation to cover the search space. The process begins by selecting an individual randomly from the population (search space) and applying mutations to the single individual, expanding the population. The parent and all the offspring (mutations) are considered for the next generation, with the fittest individual from the family surviving. At each generation the age of individual is increased, however when the offspring are determined to be the fittest and selected for survival, they inherit the age of the parent. The mutations are performed by scalar adjustment to each of the coordinates in each direction. This means that we perform $2N$ mutations in an N-dimensional search space, keeping within any bounds that may limit the search space.

As the age of the population increases, the magnitude of the mutations proportionately decreases allowing convergence toward the local extrema, and a more thorough exploration near the local extrema as the population ages. While a variety of heuristics may be used to determine the magnitude of the scalar adjustment, we use a logarithmic halving of the bounded dimensions of the search space. This results in an upper bound of $O(\log D)$ generations where D is the largest range within the search parameters. Furthermore, in an N-dimensional search space, there are N generations considered as the mutations adjust only a single parameter at a time. Finally, because each generation will perform the fitness evaluation $2N$ times, we have an upper bound of $2N^2 \log(D)$ function evaluations and an upper bound of $O(N^2 \log D)$ fitness function evaluations. Within the scope of camera calibration, we have an upper bound of the search space being five-dimensional and a reasonable practical range for the parameter space, limiting D allowing us to determine a concrete upper bound on the time complexity for camera calibration.

Mutating coordinate frames

A static coordinate frame results in premature cessation at a local extrema (the foothill problem) because the hill climber cannot move in the direction necessary to reach the true extrema. For example, if a hill climber can move in only 4 directions, say the major compass directions, when a true extrema can be reached by moving in a northwest direction the classical hill climber will fail. DHC addresses this issue by allowing a mutating (dynamic) coordinate system. DHC keeps a historical record of previous movements and constructs a new basis via a Gram Schmidt orthogonalization of the last two positions. By doing this, DHC is able to adjust for directional changes within the structure of the search space, which avoids the foothill problem in certain cases.

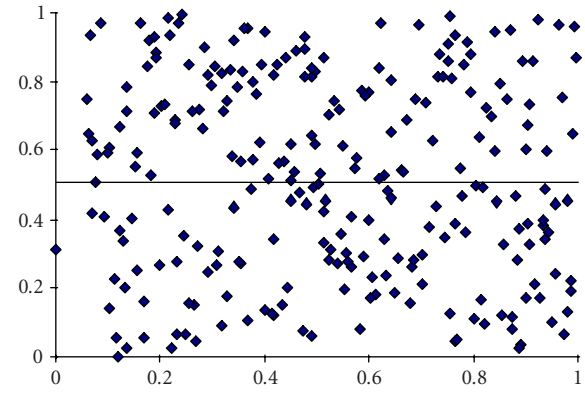

Figure 2: Scatter plot of 2D search space generated by 250 SDRS points with a trend line indicating an even disbursement of start points.

Exploiting local optima

Dynamic hill climbing also tries to avoid early convergence to a local extrema by ensuring that diversity of the population is considered directly, and independently of fitness function. Because the local hill climber has a mutation size that decreases with age, the local area is searched more thoroughly to help ensure that there is no other local extrema with better fitness. Once a local extrema is found, the individual is moved to a separate pool of static individuals that have found local extrema. When the search system stalls, DHC will examine the pool of static individuals who have achieved a local extrema and select a new population that is as different as possible from the static pool.

To facilitate this, DHC examines the hamming distance (the number of differing bits) between the two individuals and tries to maximize the distance. We note here that it is possible that this strategy is not without its own problems. The following example illustrates this. Suppose a local extrema exists at 127, bit set 11111110, the maximum hamming distance results in bit set 00000001, or 128, which is not sufficiently far from 127. However, it should be noted that a sufficiently large population reduces the probability of getting stuck when using this strategy of exploiting the local optima.

4.2. Coverage of search space

The basic idea in the DHC approach is to repeatedly perform gradient descent in the search space but to start the gradient descent in an area of the search space that is as far removed as possible from previous solutions. We call this principal of operation statistically distributed randomized starting (SDRS).

The effect is to cover the search space very thoroughly, and at the same time avoiding areas that have been previously explored and therefore avoiding the local minimum. This covers the search space very effectively, as is shown in Figure 2. In this figure we show the start points of the gradient descent in a 2D SRDS process. It is clear from the distribution that the search space is uniformly explored.

```
SDRS ( )
For each parameter in the search space,
    find the largest region that has not had a start point,
    compute a random point X in this region,
    set point X to the start point for this parameter
Endfor
Return N-dimensional StartPoint for
the next gradient descent (DHC)
```

Algorithm 1

SRDS covers the search space as completely as possible with a user specified number of starting points. Essentially SRDS is a simplified variation of DHC exploitation of local optima. The only operating parameter is the number of repeated gradient descents to try, and this is manually set to be approximately one hundred. It is important to note that the range of the calibration parameters, focal length and aspect ratio is bounded. In practice, the focal length is in the range of 1 to 5000 pixels, and the aspect ratio is in the range of .5 to 2.0. Under these conditions and operating parameters, the DHC algorithm has had good practical success.

The pseudocode for SDRS is presented in Algorithm 1.

4.3. Autocalibration algorithm

The algorithm ESTIMATE_K returns the calibration parameters in the matrix K that produced the minimum value from the cost function. It is based on the SRDS and the DHC algorithms described previously. As we have shown in the previous sections, the actual evaluation of the cost function for the two different autocalibration methods is very efficient and the upper bound on the number of calls to these functions is also known to be $O(N^2 \log(D))$. The equal eigenvalues approach requires only the computation of the eigenvalues of a 3×3 matrix, and for the Kru ppa approach the computation of three ratios based on the SVD of a 3×3 matrix. Furthermore, precomputing the SVD and storing them in a lookup table for use by the algorithm can further optimize the process and reduce the time required to execute the cost function. A single gradient descent of the cost function uses the Powell optimization algorithm [21], which is in turn based on repeated applications of the one-dimensional Brent method [21].

As we know the upper bound on the number of times the cost functions are called, we have an upper bound on the entire process of $O(N^2 \log D)$, which is the upper bound for the DHC algorithm. The remainder of the autocalibration algorithm is simply the addition of constants affecting the computation time, which are equal to the time required to execute 1 instance of the cost function. To be precise, given an image sequence of M images, and computing N intrinsic parameters, bounded by a maximum range of D, the running time on the autocalibration will be no more than $O(MN^2 \log(D))$ computations of the cost function. As we can see this is linear with respect to the number of images, as opposed to the exponential number of equations generated using the modulus-constraint-based methods.

```
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
```

Algorithm 2

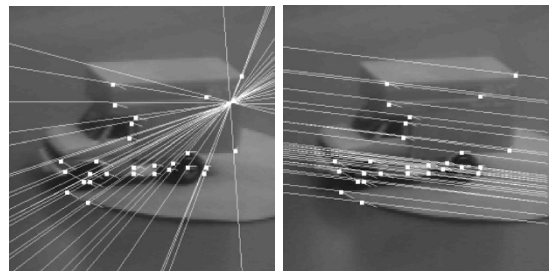

Figure 3: Two epipolar geometries that support a feature match set, yet only one can be correct [32].

The basic pseudocode for estimating K is presented in Algorithm 2.

4.4. Degeneracy

The method presented makes use of all the computed interframe geometries; however no consideration is given for incorrectly computed fundamental matrices. An incorrect fundamental matrix can occur and is known as a degeneracy case. It is commonly known that there are degenerate situations where many epipolar geometries will support the same feature match set [32].

As shown in Figure 3, we have 27 corresponding points and two computed epipolar geometries that support them. Clearly, there can be only one truly correct geometry; however, it simply takes a single outlier to potentially produce an incorrect geometry. Clearly, an incorrect fundamental matrix will result in an incorrect self-calibration when using only the one incorrect fundamental matrix.

The potential for computation of a single degenerate fundamental matrix from a sequence of images when using a RANSAC method is unavoidable and thus all computed geometries from an image sequence are to be considered. By simply using the fundamental matrix with the highest support, we will achieve incorrect results when that computed geometry is degenerate. By using all of the computed fundamental matrices, we have some knowledge of the effect each fundamental matrix has on the cost function. If we assume for demonstrations sake that we have equal confidence in each and every fundamental matrix that has been computed for an $M + 1$ image sequence, a single degenerate geometry will weigh in at $1/M$ and therefore only affect the computation proportionally to the number of images in the sequence.

Table 1: Results of autocalibration for focal length (pixels) versus other algorithms. Correspondences are computed automatically.

Name	Number of images	Stated focal	Computed focal length (equal eigenvalues)	Error % versus stated	Computed focal length (Kruppa)	Error % versus 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
ETL-Eushiba	5	837	857.25	2.4	1233.85	47.4

Handling degeneracy

While methods exist that attempt to detect degenerate configurations [33], we have chosen to use the number of supporting matches for each fundamental matrix as a measure of confidence. This metric, while not theoretically as reliable as a method that detects degeneracy, is suitable because the automated methods for computing the fundamental matrix [34] provide a relatively large number of matches with the associated fundamental matrix. Our experiments are performed under the assumption that the number of feature matches used to compute the fundamental matrix reduces the likelihood of computing a degenerate geometry. We rely on the effectiveness of the software presented in [34] to produce many feature matches and compute fundamental matrices with sufficient support that the probability of outlier caused degeneracy is greatly reduced, yet any reliable computation of the fundamental matrix will have the same result. Therefore, we use magnitude of the support feature set that was used to compute the geometry as a measure of our confidence.

Degeneracy can also be effectively handled in other ways and we outline a couple of methods next. The first obvious solution is to use the PLUNDER algorithm (pick least undegenerate randomly) outlined by Torr in [32], however it is more complicated to implement than other solutions. The benefit of handling degeneracy this way is that we can be sure that all fundamental matrices we are using are not degenerate. Another alternative is to prune fundamental matrices that produce calibrations parameters that are not consistent with the entire set. Effectively we perform a single image pair calibration for each fundamental matrix in the sequence and then perform a statistical analysis of the individual results. We can now prune any fundamental matrix whose individual calibration results are outside an acceptable level of error. Using covariance analysis or Frobenius norm will provide reasonable results.

5. EXPERIMENTAL RESULTS

There is no practical reason to autocalibrate all five intrinsic parameters [16], however, by assuming the principal point and the skew are fixed, results are encouraging. This problem is not unique to our method, and occurs in class B algorithms as well [8]. In [8], the principal point could not be computed accurately using class B algorithm, and for this reason it was also assumed to be fixed.

For many autocalibration algorithms, the evaluation of performance consists of a simple visual inspection of the resulting 3D reconstruction. This is not an adequate metric because it has been shown that the quality of the final reconstruction is visually acceptable for a wide variety of calibration parameters [16]. In order to test the capabilities of the presented evolutionary method, we used test data for which the ground truth was known; that is, the intrinsic parameters are already known a priori. Some of these data sets are the same ones used in the literature, in particular those for class B algorithms. The conclusions are that the results of class A algorithms using the evolutionary approach is comparable to that of class B algorithms, yet the simplicity and efficiency of the evolutionary method is significant. The experimental results also give an indication of what the autocalibration errors are for a typical image sequence. We performed these experiments a number of times to make sure that the results of the SRDS algorithm are repeatable and unbiased.

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 [3, 5, 7, 35]. In particular, the castle sequence [7] is used as a test case for comparison with the class B approach that requires a projective reconstruction. We see that our autocalibration results are comparable to those of other class B self-calibration algorithms.

In Table 1 we list our autocalibration results compared to the previously published results in the literature, which we assume to be correct. In the last example from [35] shown in Table 1, the error with the Kruppa autocalibration is quite large. A possible explanation is that the motion is close to being a pure translation, which is known to be a degenerate motion for the Kruppa algorithm [14, 15]. It is also a good indicator of how the equal eigenvalues method performs well in spite of these degenerate motions. In these experiments we take the image sequences as input and compute the matching feature points automatically, using the software described in [34]. In other words, we are not given matching 2D feature points, but simply a set of images. Therefore the closeness of our results to those published in the literature is significant because we are actually using different software to compute the fundamental matrices. We are also unable to verify independently that the published ground truth focal lengths are correct; it is possible that the stated focal lengths have some level of error in them as well.

Table 2: Results of autocalibration for focal length (mm) for photogrammetric sequences. Reprojection error is in pixels. Correspondences are selected by hand.

Name	Number of images	True focal length	Eigen focal length	Error %	Kruppa focal length	Error %	Correct reprojection	Eigen reprojection	Kruppa reprojection
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 3: Results of autocalibration for focal length (mm) and aspect ratio for photogrammetric sequences using the equal eigenvalue method.

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

Table 4: Results of autocalibration for focal length (mm) and aspect ratio for photogrammetric sequences using the Kruppa autocalibration method.

Name	True aspect	Kruppa aspect	Variance	Error %	True Focal	Kruppa focal	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

In the next 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 project parameters of the photogrammetric package [36]. We therefore assume that all the intrinsic parameters are set a priori, except for the focal length which we autocalibrate. Table 2 shows the autocalibrated focal length in millimeters versus the true focal length, along with the error percentage for both autocalibration methods. Since we have the associated 3D reconstructions for the corresponding 2D features, we can use more sophisticated performance measures, namely, reprojection error.

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; this independently verifies the work of Bougnoux [16].

In the next experiment we attempt to autocalibrate both aspect ratio and focal length using the two class A 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.

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 auto calibrate all parameters. In particular, the error percentage in the focal length increases slightly.

One possible explanation is that the gradient descent algorithm is stuck in a local minimum. 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 Tables 3 and 4 for the autocalibrated aspect ratio and focal length is very small over these runs. This indicates that it is highly likely that the stochastic optimization algorithm is consistently finding a local minimum that is hopefully also the global minimum.

The next set of experiments, shown in Tables 5, 6, and 7, have as input image sequences that were taken with the same camera with invariant intrinsic parameters. There are image sequences that we have taken by hand, for which ground truth is known, or from various other modeling projects (ISPRS Working Group V/2 on scene Modeling and Virtual Reality; http://www.vit.iit.nrc.ca/elhakim/WGV2-data.html). In these experiments, we again compute the correspondences automatically using the software described in [20].

Table 5: Results for autocalibration of focal length for three sequences taken from the same uncalibrated camera.

Name	Number of images	Eigen focal	Kruppa focal
Chapel	12	27.82	31.31
Climber	13	27.91	33.88
Workshop	8	26.19	38.09

Table 6: Results for autocalibration of focal length for three sequences used by the ISPRS Working Group on Scene Modeling and Virtual Reality.

Name	Number of images	Eigen focal	Kruppa focal
Indoor	5	1663	1815
Waterways	3	1759	fail
Building	2	1609	fail

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 3.42 mm. It is not surprising that the autocalibration results differ, since certain motions are degenerate with regards to the Kruppa-based autocalibration [14]. 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. Furthermore, longer sequences converge with a more accurate estimation of the intrinsic camera parameters.

The final set of experiments, shown in Tables 6 and 7, has as input image sequences that are used as test data for the ISPRS Working Group *V*/2 on Scene Modeling and Virtual Reality. These images are used to test different model building software packages, and the ground truth is known. In Tables 6 and 7, we again compute the correspondences automatically using the software described in [20], and autocalibrate only the focal length. We see in Table 6 that the results are reasonable given that the true focal length is 1737 pixels in all cases, but that sometimes Kruppa's approach does not converge. The likely causes are sensitivity to motion degeneracy and the difficulty of convergence with a small number of images associated with the Kruppa method.

Table 7 presents a variety of experiments also from the ISPRS workgroup. In certain examples that error is very large, however the average error is only 17.25% with a standard deviation of 21.99. By removing the two grossly incorrect samples from the table, the error percent and standard deviation dropped by almost half to 9.54 and 12.11, respectively.

In summary, Table 1 shows that the evolutionary approach is as good as the published results for class B algorithms, particularly the castle sequence. However, class B algorithms are not easily scalable from a computational point of view, and thus cannot handle long image sequences. Class A, fundamental matrix-based, approaches are computationally very efficient because single evaluations of the cost functions do not take long and accuracy increases as the sequence length increases. The time taken for autocalibration is in the order of seconds for all the image sequences on a 400 MHz Pentium II processor. It seems that the equal eigenvalues method is superior to Kruppa's method for degenerate motions and smaller sets of images. There are cases, however, where Kruppa's method clearly outperforms the equal eigenvalues method. Further investigation is necessary to determine whether or not a heuristic can be developed to choose one algorithm over the other by predetermining the camera motion using arbitrary intrinsic camera parameters in a first step and using this knowledge to select an appropriate class A or class C algorithm that uses an evolutionary approach.

6. CONCLUSIONS

This work presents an algorithm for self-calibration that has four major advantages:

(1) simplicity (and ease of implementation),
(2) accuracy and reliability,
(3) scalability (handles very long sequences),
(4) speed of execution (known upper bound).

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 [14, 15, 23]. However, we show that for nondegenerate motions both methods perform equally well when we are calibrating only the focal length, or the focal length and aspect ratio. The equal eigenvalues approach, combined with evolutionary methods is very simple and performs as well as any class B method we compared it against. While it is theoretically equivalent to the Kruppa approach, it performs better numerically in situations where we are closer to degenerate motions, such as pure translation, and seems to converge better for smaller sets of images. Experimentally we have shown that evolutionary-based autocalibration using class A algorithms produces similar results to their class B counterparts.

We have shown that in practice the statistically distributed random starting (SDRS) helps to reliably find a consistent local minimum of the cost function that we expect to be the global minimum. We have also shown that the error in the autocalibration of the focal length is usually in the range of 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 currently require exact depth information.

When dealing with long image sequences, class B algorithms will produce a set of polynomial equations for each image pair. This results in a large system of equations for the entire image sequence. Continuation methods can solve small systems of equations but are ill posed when the number of equations becomes large. The methods proposed in this work have advantages for long image sequences. The methods we have described are computationally efficient with a known upper bound that is better than any published class

Table 7: Results for autocalibration of focal length and comparison to ground truth.

Project	Focal length (in pixels)	Computed focal length (in pixels)	Number of images	Error %
Amsterdam	1736.7	1866.7	4	7.48
Benches	1736.7	612	7	64.76
Chapel-l	2105	1640	2	22.0
Chapel-S	2105	1473	7	30.0
Corfu	2923.4	2995	7	0.02
Fitting	1684	1681	2	0.001
Florence	1897.3	1787	6	0.058
Light	2348.3	3647	4	55
Nikh	2348.3	2348	2	0.001
Oldbuild	1649.5	1588	7	0.037
Reg-1	2095	1609	7	23.1
Reg-2	611	747	27	22.2
Sphinx	1754	1764	16	0.0057

B method on long image sequences and produces comparable results. It is also the case that processing long image sequences is advantageous in that any error for an individual fundamental matrix (e.g., because of a degenerate motion) will have less of an impact on the final result. For example, an M image sequence has $M - 1$ adjacent pairs and therefore $M - 1$ representative fundamental matrices. As M becomes larger (i.e., the number of images in the sequence increases), the individual error associated with a single image pair has less effect. The accuracy of the estimation increases only with the size of the image sequence. As the sequence length tends to infinity, the error can be more closely associated to the error within the individual computation of the fundamental matrix. Another advantage of long image sequences is that the global optimum is better defined than when using short image sequences. In other words, with long sequences the global optimum tends to be sharper and better defined making the results more stable.

Due to a lack of standardized data sets that can be used to effectively benchmark different autocalibration routines, the "look" of a resulting reconstruction is often used as a benchmark, which is not appropriate for performance evaluation. For proper performance analysis of autocalibration algorithms, it would be very useful to have a standardized set of images for which the ground truth is known. A start has been made by ISPRS Working Group, but more needs to be done. At the very least, results of using such test data should include the accuracy of the parameter values, consistency of results (similar to Table 4), and an accuracy of image sequence length ratio benchmark.

Evolutionary-based autocalibration with varying intrinsic parameters still remains an open problem, however it is conceivable to adapt the cost functions to allow for varying focal lengths between image pairs.

REFERENCES

[1] R. Hartley, "Kruppa's equations derived from the fundamental matrix," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 2, pp. 133–135, 1997.

[2] 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.

[3] L. Lourakis and R. Deriche, "Camera self-calibration using the svd of the fundamental matrix," Tech. Rep. 3748, INRIA, Sophia Antipolis, France, August 1999.

[4] P. Mendonca and R. Cipolla, "A simple technique for self-calibration," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 112–116, Fort Collins, Colo, USA, June 1999.

[5] C. Zeller and O. Faugeras, "Camera self-calibration from video sequences: the kruppa equations revisited," Tech. Rep. 2793, INRIA, Sophia Antipolis, France, February 1996.

[6] M. Pollefeys and L. Van Gool, "Stratified self-calibration with the modulus constraint," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 21, no. 8, pp. 707–724, 1999.

[7] M. Pollefeys, *Self-calibration and metric 3D reconstruction from uncalibrated image sequences*, Ph.D. thesis, Catholic University Leuven, Leuven, Belgium, 1999.

[8] M. Pollefeys, R. Koch, and L. Van 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.

[9] B. Triggs, "Autocalibration from planar scenes," in *Proc. European Conference on Computer Vision*, pp. 89–105, Freiburg, Germany, June 1998.

[10] P. Sturm and S. Maybank, "On plane-based camera calibration: A general algorithm, singularities, applications," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, vol. I, pp. 432–437, Fort Collins, Colo, USA, June 1999.

[11] B. Triggs, "Autocalibration from planar scenes," in *Proc. European Conference on Computer Vision*, pp. 89–105, Freiburg, Germany, June 1998.

[12] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[13] É. Vincent and R. Laganière, "Detecting planar homographies in an image pair," in *Proc. 2nd International Symposium on Image and Signal Processing and Analysis*, pp. 182–187, Pula, Croatia, June 2001.

[14] O. Faugeras and Q.-T. Luong, *The Geometry of Multiple Images*, MIT Press, Cambridge, Mass, USA, 2001.

[15] P. Sturm, "A case against Kruppa's equations for camera self-calibration," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1199–1204, 2000.

[16] S. Bougnoux, "From projective to Euclidean space under any practical situation, a criticism of self-calibration," in *Proc. 6th International Conference on Computer Vision*, pp. 790–796, Bombay, India, 1998.

[17] S. Kirkpatrick, C. Gelatt Jr., and M. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671–679, 1983.

[18] A. Whitehead and G. Roth, "Evolutionary based autocalibration from the fundamental matrix," in *EvolASP Workshop Proceedings*, pp. 292–303, Springer-Verlag, Kinsale, Ireland, April 2002.

[19] A. Morgan, *Solving Polynomial Systems Using Continuation for Engineering and Scientific Problems*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1987.

[20] A. Fusiello, "Uncalibrated Euclidean reconstruction: a review," *Image and Vision Computing*, vol. 18, no. 6-7, pp. 555–563, 2000.

[21] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C*, Cambridge University Press, Cambridge, UK, 1988.

[22] M. de la Maza and D. Yuret, "Dynamic hill climbing," *AI Expert*, vol. 9, no. 3, pp. 27–31, 1994.

[23] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, Cambridge, UK, 2000.

[24] Q.-T. Luong, R. Deriche, O. D. Faugeras, and T. Papadopoulo, "On determining the fundamental matrix: analysis of different methods and experimental results," Tech. Rep. RR-1894, INRIA, Sophia Antipolis, France, 1993.

[25] Z. Zhang, "Determining the epipolar geometry and its uncertainty: a review," Tech. Rep. RR-2927, INRIA, Sophia Antipolis, France, 1996.

[26] R. Hartley, "In defense of the 8-point algorithm," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 6, pp. 580–593, 1997.

[27] H. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature*, vol. 293, no. 10, pp. 133–135, 1981.

[28] K. W. Nicholson, *Linear Algebra with Applications*, PWS Publishing Company, Boston, Mass, USA, 3rd edition, 1995.

[29] J. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, Mich, USA, 1975.

[30] K. De Jong, *An analysis of the behavior of a class of genetic adaptive systems*, Ph. D. dissertation, University of Michigan, Ann Arbor, Mich, USA, 1975.

[31] D. Laurendeau, G. Roth, and L. Borgeat, "Optimization algorithms for range image registration," in *Vision Interface*, pp. 141–151, Toronto, Canada, May 1996.

[32] P. Torr, A. Zisserman, and S. Maybank, "Robust detection of degenerate configurations for the fundamental matrix," in *Proc. 5th International Conference on Computer Vision*, pp. 1037–1042, Boston, Mass, USA, June 1995.

[33] P. Gurdjos and P. Sturm, "Methods and geometry for plane-base self-calibration," in *Proc. International Conference on Computer Vision and Pattern Recognition*, Madison, Wis, USA, June 2003.

[34] A. Whitehead and G. Roth, "The projective vision toolkit," in *Proceedings Modeling and Simulation*, pp. 204–209, Pittsburgh, Pa, USA, May 2000.

[35] T. Ueshiba and F. Tomita, "A factorization method for projective and Euclidean reconstruction," in *Proc. 5th European Conference on Computer Vision*, vol. I, pp. 290–310, Springer-Verlag, Freiburg, Germany, June 1998.

[36] Photomodeler by EOS Systems Inc., http://www.photomodeler.com.

Anthony Whitehead received a Bachelor of Computer Science (BCS) from Carleton University in 1996. He is currently completing his Ph.D. His main research interests include computer vision, image/video indexing and retrieval, video/image processing, and biometrics.

Gerhard Roth received a Bachelor of Mathematics from the University of Waterloo in 1976, a Master's of computer science from Carleton University in 1984, and a Ph.D. in electrical engineering from McGill in 1993. He has done research in evolutionary algorithms, robust statistics, and model building from 3D data. His current research area is the field of computational video.