

Autocalibration method for scanning electron microscope using affine camera model

Andrey Kudryavtsev, Valerian Guelpa, Patrick Rougeot, Olivier Lehmann, Sounkalo Dembele, Peter Sturm, Nadine Piat

▶ **To cite this version:**

Andrey Kudryavtsev, Valerian Guelpa, Patrick Rougeot, Olivier Lehmann, Sounkalo Dembele, et al.. Autocalibration method for scanning electron microscope using affine camera model. Machine Vision and Applications, 2020, 31 (7), pp.1 -15. 10.1007/s00138-020-01109-x . hal-02993826

HAL Id: hal-02993826
https://hal.science/hal-02993826

Submitted on 7 Nov 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Noname manuscript No.
(will be inserted by the editor)

Autocalibration Method for Scanning Electron Microscope using Affine Camera Model

Andrey V. Kudryavtsev · Valérian Guelpa · Patrick Rougeot · Olivier Lehmann · Sounkalo Dembélé · Peter Sturm · Nadine Le Fort-Piat

Received: date / Accepted: date

Abstract This paper deals with the task of autocalibration of Scanning Electron Microscope (SEM), which is a technique allowing to compute camera motion and intrinsic parameters. In contrast to classical calibration, which implies the use of a calibration object and is known to be a tedious and rigid operation, auto- or selfcalibration is performed directly on the images acquired for the visual task. As autocalibration represents an optimization problem, all the steps contributing to the success of the algorithm are presented: formulation of the cost function incorporating metric constraints, definition of bounds, regularization, and optimization algorithm.

The presented method allows full estimation of camera matrices for all views in the sequence. It was validated on virtual images as well as on real SEM images (pollen grains, cutting tools, etc.). The results show a good convergence range and low execution time, notably compared to classical methods, and even more in the context of the calibration of SEM.

Keywords Autocalibration · SEM · Affine camera · Global optimization

1 Introduction

The use of vision in robotics currently requires camera calibration, *i.e.* the knowledge of its motion (extrinsic parameters) and model (intrinsic parameters). In the present work, the aim is to achieve calibration of a Scanning Electron Microscope (SEM) directly from images of a specimen (like pollen grains presented in Fig. 1).

For SEM, the perspective effects can be neglected and the affine camera model is considered [1,2]. But even if this subject is well studied, the calibration can be very complex due to the following reasons. First, in most cases it requires a calibration object. It often means a special step of fabrication, which can be very expensive and time consuming especially when working with SEM as, at microscale, the calibration object is of micrometric size. Moreover, it is very difficult to guarantee the quality of its fabrication, which has a profound impact on the accuracy of further image processing. Secondly, the classic calibration [3] needs to be done offline, which can be very restrictive in some applications where the calibration object can't be placed in front of the camera once the operation started. Thirdly, which includes partially the second point, there is a problem of maintainability of calibration parameters. In order to re-calibrate a camera, the main operation task has to be stopped.

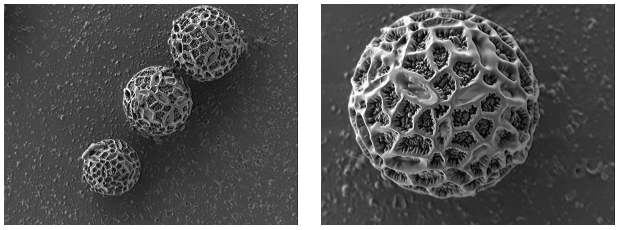


Fig. 1 Examples of SEM images with 100 µm pollen grains.

It is important to mention that computation of intrinsic camera parameters is inseparable from camera motion, i.e. extrinsic camera parameters. In fact, autocalibration is a method of calibration, allowing to recover both the intrinsic and extrinsic camera parameters, and is carried out using the same images required for performing the visual task [4]. Most autocalibration methods use a projective calibration as a starting point: it is a calibration from point correspondances only, without any knowledge about camera and scene (scale factor, aspect ratio, skew between image axis, scene orientation, position of principal point, etc.), and it is different from the metric, i.e. true, one up to a 3D projective

Address(es) of author(s) should be given

transformation. Affine calibration and euclidean calibration differs from metric one up to an affine transformation and a similarity transformation, respectively. In the literature, Euclidean calibration is often referenced as similarity or metric calibration [5]. Furthermore, in this work, the term metric will be referred only to a true calibration with the known scale. In our case, as there is no particular information about the object size only Euclidean calibration is achievable, i.e., an up-to-scale calibration. There is a variety of methods for projective calibration computation and the most common ones are factorization-based methods [6, 7]. Then, once the projective calibration is obtained, the goal of autocalibration algorithms is to determine a rectifying homography **H** from autocalibration constraints and then to transform the calibration to an Euclidean one. However, these methods are often blamed for instability [8].

In case of affine camera, by using a factorization technique, it is possible to directly obtain an affine calibration [9]. It means that the plane at infinity is already in its canonical position and the goal is hence to determine the intrinsic parameters of the camera in order to upgrade the calibration from affine to Euclidean. A variety of methods are based on the use of some calibration constraints (zero skew, known aspect ratio, etc.) or motion constraints (pure rotation, planar motion, etc.) [10–12]. One of the most significant article in the field of autocalibration of affine camera is the work of *Long Quan* [13]. In his paper, the author proposes a method allowing to upgrade the affine calibration (in this case obtained with a factorization algorithm) to the Euclidean one by using an optimization algorithm. The goal is to find a non-singular 3×3 homography matrix and the criterion of optimization is based on Euclidean motion constraints. One of the steps of estimation contains a Cholesky decomposition, which accepts only positive-definite matrices. A priori, this condition is not satisfied in the presence of noise in real images [14]. Thus, the author proposed an elegant way to avoid this problem by imposing the constraints on the matrix to decompose and to assure by this that it is positive-definite. However, the experiments that will be presented further have shown that the method of *Long Quan* (further referenced as LQ) can not guarantee the respect of the metric constraints, such as aspect ratio close to one, at least in case of SEM.

As many other selfcalibration algorithms, LQ uses local optimization that aims to find a local minimum of the objective function, *i.e.* the minimum that is the closest to the initial solution. However, it is not sufficient for the application presented here, because the objective function contains non-linearities which results in non-convex cost function. It means that one should use more complex techniques allowing to expand the search space. More recently, this problem was addressed by the methods of global optimization. *Fusiello et al.* in [15] addressed this problem by an interval branch-and-bound method employed for numerical minimization based on constraint on the fundamental matrix. In [16], the global optimization was used to compute the dual image of the absolute conic to find a rectifying homography. However, even if these global optimization algorithms can guarantee a theoretical global optimality, their lock is a computational time which turns out to be a critical issue for some applications. According to *Heinrich et al.*, the fundamental limitation of these algorithms is that the minimized objective function has no particular geometric meaning which results in instability of these methods [17]. In return, they proposed a method based on a maximum likelihood objective function.

This paper presents a new method of autocalibration of SEM, considered as an affine camera, which allows to directly obtain Euclidean set of camera matrices by means of global optimization, without passing by affine calibration. It is worth mentioning that it is considered that camera intrinsic parameters are constant and the scene is rigid, i.e. the change of 2D projections across the images is due only to the camera motion. The crucial topics contributing to the success of the algorithm are discussed. Section 2 introduces the specificities of Scanning Electron Microscope. Section 3 presents the representation of SEM considered as an affine camera. The formulation of optimization criteria that include error function, bound constraints and regularization is presented in Section 4. Section 5 is about the global optimization algorithms. Next, the robustness and convergence range of the auto-calibration is tested on synthetic images and compared to classical methods in Section 6. Finally the auto-calibration is tested on real SEM images in Section 7.

2 Background of SEM

From a physical point of view, electronic imaging is based on the interaction between a radiation of high energy electrons and the atoms of matter. Kinetic energy of an electron accelerated with the voltage **E** is **eE**, where **e** is the charge of the electron. The relation of *De Broglie* shows that the wavelength λ of the electron is inversely proportional to its level of energy, and then to its accelerating voltage **E** [18]. This leads to very low values of wavelength compared to light (radiation of photons), in the range of X-rays radiation. According to *Abbe* formula, the resolving power of a radiation (the minimal distance required to resolve two objects) is about $\lambda/2$, then with electrons the resolving power can be very low, i.e. the imaging resolution very high: for 1 V accelerating voltage, the resolving power is about 1.2 nm that is about 200 times smaller than optical imaging resolving power; for 2 kV accelerating voltage, the resolving power is about 0.02 nm that is about 40 000 times smaller than optical imaging resolving power [18] [19]. Finally electron imaging

is a very high resolution imaging, up to atomic scale. Nowadays, SEM is a central observation instrument for material and life sciences, and micro-nano-engineering [20].

SEM has the following working principle. High energy electrons, that collide with the atoms of a sample, induce two types of scattering: elastic and inelastic [21]. Elastic scattering comes with elastic shock, usually with kernel: there is no energy loss, instead, electron is diffracted and atomic bonds are broken. After the shock, the electron becomes a Back-Scattered Electron (BSE). Inelastic scattering comes with inelastic shock, usually with an electron of the cloud: there is a loss of energy. After the collision, the atom is ionized and emits a Secondary Electron or SE. The scanning of the sample surface and the collection of back-scattered or secondary electrons leads to a 2D image of that surface, reflecting the atomic number and topography, respectively. Fig. 1 shows some examples of pollen grains images from secondary electrons detector: the topography is very well presented.

A Scanning Electron Microscope comprises at least the following components (Fig.2):

1. a gun (source) to generate the beam of electrons,
2. a condenser lens to make the beam cylindrical as possible,
3. an aperture and astigmatism lenses to reduce the diameter of the beam,
4. an objective lens to focus the beam onto the specimen,
5. some scanning lenses to scan the surface of the sample by the beam,
6. a detector to collect and amplify the emitted electrons and to generate pixels.

All these components are located inside a vacuum environment in order to avoid every possible interaction between gas molecules and electrons (both emitted and detected).

For the experiments, in this project, we used a Carl Zeiss Auriga FE-SEM along with its own computer, and a remote computer to run user applications (Fig. 3). The SEM, that is equipped with a six degrees-of-freedom micro-manipulator for object manipulation, features an electron column based on Schottky field emission gun and three detectors (SE detector in the chamber, SE detector in the column, BSE detector in the column). We used the SE detector in the chamber. Some specifications of the SEM are: magnification ranging from $\times 10$ to $\times 10^6$, working distance from 0 to 20 mm, maximum frame rate of 20 Hz.

3 Geometrical modeling of SEM

While SEM has no matrix sensor, it still can be modeled as a camera. According to the literature [22–24], in case of SEM,

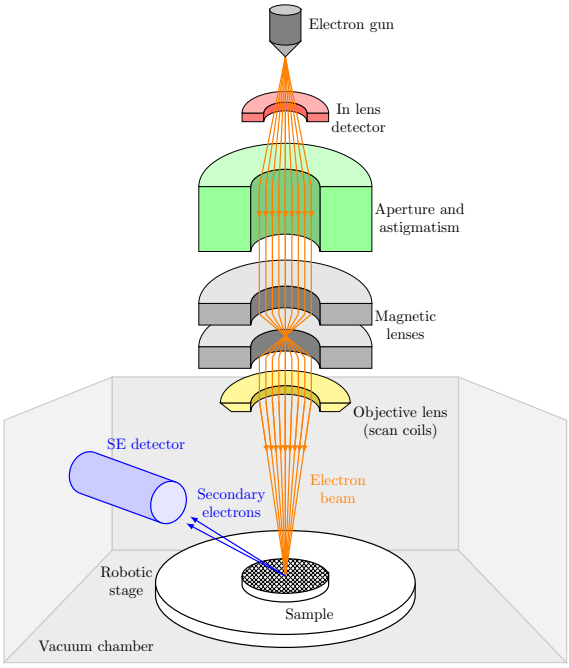


Fig. 2 Basic SEM internal structure.



Fig. 3 Carl Zeiss Auriga SEM used in experiments.

the perspective effects can be neglected for magnification values bigger than $\times 1000$ and then an affine model may be used.

Affine camera model assumes that all projection rays are parallel to each other, *i.e.* the camera centre is at infinity. That is why affine camera model is often referred as parallel projection model. With regard to SEM, consider the following situation which is quite common: the object with a size of 10 µm is visualized using SEM tuned to the working distance of 10 mm. The ratio between the distance camera-object and the object size is equal to 1000. Evidently, in such conditions, the projection rays are very close to be parallel. This example gives the intuition behind the usage of affine camera model for SEM. Getting back to geometry, the pro-

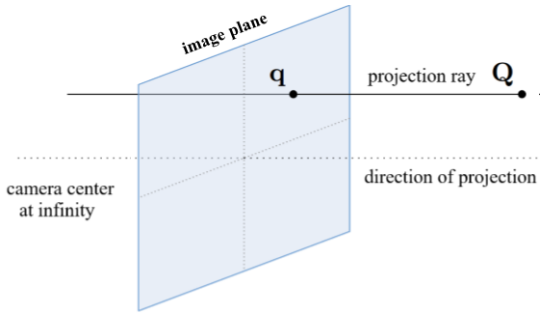


Fig. 4 Image formation under assumption of affine camera model.

cess of image formation with an affine camera is represented schematically in Fig. 4.

The camera matrix $\mathbf{P}$ in this case is different from perspective projection by a parallel projection matrix Π:

$$\mathbf{P} = \mathbf{K}\Pi \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \tag{1}$$

where $\mathbf{K}$ is the matrix of intrinsic parameters, $\begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix}$ is the motion matrix ($\mathbf{R}$ rotation matrix, $\mathbf{t}$ translation vector) and $\Pi = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$.

As a result, the affine projection matrix is of the form:

$$\mathbf{P} = \begin{pmatrix} * & * & * & * \\ * & * & * & * \\ 0 & 0 & 0 & * \end{pmatrix} \tag{2}$$

Affine camera does not have principal point as all projection rays are parallel (camera center is infinitely far from the image plane). It has a direct impact on the form of intrinsic matrix $\mathbf{K}$ that has the following form in affine case:

$$\mathbf{K} = \begin{pmatrix} \alpha f & s' & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} = diag(f, f, 1) \begin{pmatrix} \alpha & s & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3}$$

with the three following parameters:

- Aspect ratio α that represents the ratio between height and width of a pixel. In ideal situation its value is equal or at least close to one. It is also true for SEM images which was confirmed in [24].
- Skew parameter s reflects the orthogonality level of $\mathbf{x}$ and $\mathbf{y}$ axis in image frame. Its value should be close to zero.
- Overall scale factor f represents $\frac{pixel}{m}$ ratio.

In addition to the special form of the intrinsic parameters matrix, parallel projection imposes some other interesting properties. First, image is invariant to the object displacement along the direction of projection (see Fig.5.a). In other words, image is invariant to the distance between camera and object, which is the case in SEM. For instance, moving the sample closer to the electron beam or moving it away will not change the resulting 2D projection. As a result, the depth coordinate is completely lost in the process of image formation in case of parallel projection.

Secondly, the projection of the object is independent of translations in x and y directions of image frame if the relative coordinates are used both in 3D object frame and in camera frame (see [13]). If such translation is performed, only the position of the object in the image changes, but not the relative disposition of its feature points (see Fig.5.b). A reference point is often chosen as the centroid of 2D projection which is the centroid of corresponding 3D points at the same time. The exact same points may be obtained by using relative coordinates and translating the centroid to $(0,0)^\top$.

In the case of SEM the displacement of 2D projections is due to the object motion (the motion of the robotic stage). However, for the remainder of the manuscript we will consider that the camera is moving and the object is fixed. Both situations are geometrically equivalent, yet, the latter one simplifies a lot the presentation.

Concerning the subject of motion estimation, it is known that motion parameters as well as the camera intrinsic parameters are contained in the camera matrix which has a special form in affine case (see Eq. 2) [14]. It can be decomposed as:

$$\mathbf{P} = \underbrace{\begin{pmatrix} \alpha f & s & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\mathbf{K}} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ 0\,0\,0 & 1 \end{pmatrix} \tag{4}$$

$$= \underbrace{\begin{pmatrix} \mathbf{A}_{2\times 2} & 0_{2\times 1} \\ 0 & 1 \end{pmatrix}}_{\mathbf{K}} \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{5}$$

Thus, the 2D projection $\mathbf{q}$ of a 3D point $\mathbf{Q}$ is written as (in homogeneous coordinates):

$$\mathbf{q} = \mathbf{P}\mathbf{Q} \tag{6}$$

which is equivalent to

$$\begin{pmatrix} q_x \\ q_y \end{pmatrix} = \begin{pmatrix} \alpha f & s \\ 0 & f \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \end{pmatrix} \begin{pmatrix} Q_x \\ Q_y \\ Q_z \end{pmatrix} + \begin{pmatrix} \alpha f & s \\ 0 & f \end{pmatrix} \begin{pmatrix} t_x \\ t_y \end{pmatrix} \tag{7}$$

Finally, taking into account all the described properties and equations, the total number of parameters for an affine camera is: $3_{\text{intrinsic}} + 3_{\text{rotations}} + 2_{\text{translations}} = 8$.

In this way, for N_{im} images corresponding to as many camera poses, the total number of parameters is: $3_{\text{intrinsic}} + 3_{\text{rotations}}.N_{im} + 2_{\text{translations}}.N_{im} = 3 + 5.N_{im}$.

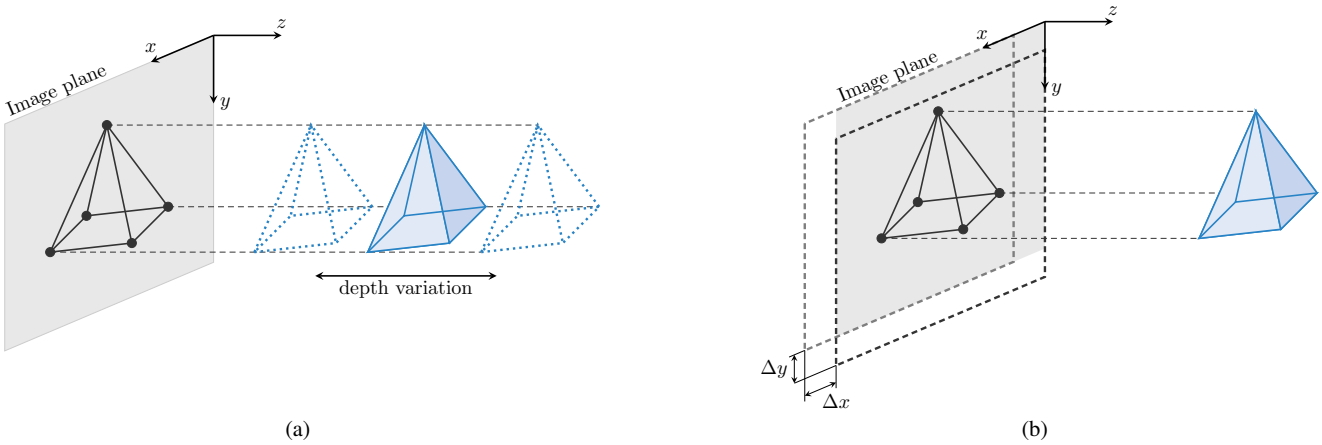


Fig. 5 Illustration of affine camera properties: a) invariance to depth variation; b) invariance to in-plane translations of camera or object.

4 Problem formulation

Now the challenge is to find a method to compute the intrinsic and extrinsic parameters of affine camera. Taking as a basis a measurement matrix, we will define a principled cost function usable in an global search optimization method.

4.1 Measurement matrix

Mathematically, a set of feature correspondences is represented as a matrix $\mathcal{W}$ that will be further referred as *measurement matrix*. Consider that N_{im} images of the same object were taken from different view points by the same but moving affine camera. It is possible to extract a measurement matrix $\mathcal{W}$ that contains the projections of N_{pt} 3D points in different images:

$$\mathcal{W} = \begin{bmatrix} \mathbf{W}_1 \\ \mathbf{W}_2 \\ \vdots \\ \mathbf{W}_{N_{im}} \end{bmatrix} = \begin{bmatrix} \mathbf{q}_1^1 & \mathbf{q}_2^1 & \cdots & \mathbf{q}_{N_{pt}}^1 \\ \mathbf{q}_1^2 & \mathbf{q}_2^2 & \cdots & \mathbf{q}_{N_{pt}}^2 \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{q}_1^{N_{im}} & \mathbf{q}_2^{N_{im}} & \cdots & \mathbf{q}_{N_{pt}}^{N_{im}} \end{bmatrix} \tag{8}$$

with $\mathbf{W}_i$ the set of features extracted from the i-th image:

$$\mathbf{W}_i = \begin{bmatrix} \mathbf{q}_1^i & \mathbf{q}_2^i & \cdots & \mathbf{q}_{N_{pt}}^i \end{bmatrix} \tag{9}$$

Several remarks are to be done about the measurement matrix:

- the number of columns is equal to the number N_{pt} of extracted features,
- one column corresponds to N_{im} projections of one 3D point,
- the number of rows is equal to $3N_{im}$ in homogeneous coordinated,
- all features are viewed in all images.

4.2 Cost function formulation

The problem of autocalibration can be formulated as the following optimization problem:

$$\arg\min_{\xi \in \mathbb{R}^n} f(\xi) \tag{10}$$

where ξ is a vector of camera's parameters (both intrinsic and extrinsic) and $f(\xi)$ is the cost or objective function representing the error. Obviously, the formulation of this function is a decisive point to obtain good properties for autocalibration such as robustness and accuracy.

Knowing that every element of $\mathcal{W}$ can be obtained by the multiplication of 3D point coordinates and the matrix of camera in which it is projected, the expression for the *estimated* measurement matrix ($\hat{\mathcal{W}}$) can be written as follows:

$$\hat{\mathcal{W}} = \begin{bmatrix} \hat{\mathbf{W}}_1 \\ \hat{\mathbf{W}}_2 \\ \vdots \\ \hat{\mathbf{W}}_{N_{im}} \end{bmatrix} = \begin{bmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \vdots \\ \mathbf{P}_{N_{im}} \end{bmatrix} \begin{bmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 & \cdots & \mathbf{Q}_{N_{pt}} \end{bmatrix} = \mathcal{P}\mathcal{Q} \tag{11}$$

where $\mathcal{P}$ is a stack of camera matrices ($3N_{im} \times 4$ matrix containing camera matrices for all views and $\mathcal{Q}$ is the set of 3D points in homogeneous coordinates with the size $4 \times N_{pt}$. Every camera matrix ($\mathbf{P}_i$) has 8 degrees-of-freedom. Furthermore, it will be considered that images are taken with the same camera, thus only 5 extrinsic parameters have to be estimated *separately* for every image. Thus, we need to find such $\mathcal{P}$ and $\mathcal{Q}$ that would minimize the difference between $\mathcal{W}$ and $\hat{\mathcal{W}}$. In this formulation the total number of parameters is equal to:

$$\text{length}(\xi) = \underbrace{3}_{\text{intrinsic}} + \underbrace{3N_{im}}_{\text{rotation}} + \underbrace{2N_{im}}_{\text{translation}} + \underbrace{3N_{pt}}_{\text{3D points}} \tag{12}$$

The first item in this equation stands for three intrinsic parameters (scale factor, aspect ratio and skew) of the camera;

the second and third ones represent motion parameters that are different for each view; the last component is the 3D coordinates of object points.

Next, we undertake several steps allowing to reduce the number of parameters. Actually, as general optimization problems involve more variables to be optimized, it is harder to make them well-constrained. If they are not well-constrained, the optimization will proceed in a way that minimizes the global mathematical error, but that does not correspond to a solution of a real problem, i.e. has no physical meaning. Thus, it is very important to reduce the number of parameters as much as possible.

Step 1. On the basis of only images taken, only Euclidean calibration could be achieve, and not metric one. In order to make the upgrade to metric calibration, one can use the information about the pixel size for a given magnification which is generally given by SEM manufacturer. Here the scale factor f will be considered constant and equal to one, because the metric calibration is not needed. In this way, only two intrinsic parameters have to be founded.

Step 2. As the position of the world frame is unknown, we are free to fix its orientation equal to the frame of the first camera, so that:

$$\mathbf{R}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

which allows to exclude three parameters:

$$\text{length}(\xi) = 2 + 3(N_{im} - 1) + 2N_{im} + 3N_{pt} \tag{13}$$

Step 3. The biggest term in (12) corresponds to 3D points. One of the possible ways to eliminate it is to use the pseudo-inverse of the matrix $\mathcal{P}$. This solution reduce drastically the number of parameters, in return for more calculations. Indeed, the stack of 3D points $\mathcal{Q}$ may be replaced by:

$$\mathcal{Q} = \mathcal{P}^{+}\mathcal{W} \tag{14}$$

where $\mathcal{P}^{+}$ is the pseudo-inverse of the matrix $\mathcal{P}$. Here, we obtain it using SVD (Singular Value Decomposition) algorithm. Therefore, (11) transforms into:

$$\hat{\mathcal{W}} = \mathcal{P}\mathcal{P}^{+}\mathcal{W} \tag{15}$$

and the number of parameters is reduced to:

$$\text{length}(\xi) = 2 + 3(N_{im} - 1) + 2N_{im} \tag{16}$$

Step 4. The number of parameters can be further reduced by using relative coordinates and eliminating translations. Indeed, as stated in section 3 and Fig.5.b, affine model is invariant to translations in image plan. The use of relatives coordinates will allow to eliminate $\mathbf{t}_i$ and the number of parameters to estimate can be reduced to:

$$\text{length}(\xi) = 2 + 3(N_{im} - 1) \tag{17}$$

and the equation (15) takes the following form:

$$\hat{\mathcal{W}} = \mathcal{M}\mathcal{M}^{+}\mathcal{W}_r \tag{18}$$

In this expression, $\mathcal{W}_r$ is the measurement matrix in non-homogenious relative coordinates and $\mathcal{M}$ is:

$$\mathcal{M} = \begin{bmatrix} \mathbf{M}_1 \\ \mathbf{M}_2 \\ \vdots \\ \mathbf{M}_i \end{bmatrix} = \begin{bmatrix} \mathbf{A}\mathbf{R}_1 \\ \mathbf{A}\mathbf{R}_2 \\ \vdots \\ \mathbf{A}\mathbf{R}_i \end{bmatrix} \tag{19}$$

where $\mathbf{M}$ is a 2×3 upper-left components of affine camera matrix $\mathbf{P}$.

Recall that

$$\mathbf{A} = f \begin{pmatrix} \alpha & s \\ 0 & 1 \end{pmatrix} \tag{20}$$

and $\mathbf{R}_i$ is the first two rows of the rotate matrix, according to [25]:

$$\mathbf{R}_i = \mathbf{R}_z({}^{i}\theta_{i-1})\mathbf{R}_y(\rho_{i,i-1})\mathbf{R}_z^{\top}({}^{i-1}\theta_i) \tag{21}$$

where the angles θ are in-plane rotations and ρ is out-of-plane rotation.

In-plane rotation can be estimated from a pair of affine images, but there is an issue with out-of-plane rotation estimation because of bas-relief ambiguity: from two affine views, the scene (motion and structure) may be recovered only up to the combination of depth variation and out-of-plane rotation [26].

With three or more views, this ambiguity disappears in general. In our implementation, we use the self-calibration approach of [25] which computes directly, from the affine factorization of three views, the scene (structure and pose of the three views), up to an Euclidean transformation, as well as the intrinsic parameters. We use this to initialize a global non-linear optimization of all variables, as described in the following. Note that using 3 views for this initialization step makes it straightforward and efficient to embed it in a robust estimation scheme such as RANSAC.

Ideally, we are looking for such set of parameters ξ that ensure $\hat{\mathcal{W}} = \mathcal{W}_r$. However, due to the presence of noise in the estimation of image features, this equality is never satisfied exactly, which means that a way of comparison needs to be found. As in common, a geometric distance d between estimated and measured points will be used in present work. *Hartley and Zisserman* in [14] specify that the minimization of geometric error between measured and estimated 2D points is equivalent to finding such a $\hat{\mathcal{W}}$ as close as possible to $\mathcal{W}_r$ in Frobenius norm. Thus, the autocalibration task converges to an optimization problem with the following cost function:

$$f(\xi) = \sum_{i=1}^{N_{im}} \sum_{j=1}^{N_{pt}} d(\tilde{\mathbf{q}}_j^i, \mathbf{M}_i(\xi)\mathbf{M}_i^+(\xi)\tilde{\mathbf{q}}_j^i)^2 \quad (22)$$

$$= \|\mathcal{W}_r - \hat{\mathcal{W}}\|_F^2 \quad (23)$$

$$= \|\mathcal{W}_r - \mathcal{M}(\xi)\mathcal{M}^+(\xi)\mathcal{W}_r\|_F^2 \quad (24)$$

$$= \|\mathcal{W}_r - \mathcal{M}\mathcal{M}^+\mathcal{W}_r\|_F^2 \quad (25)$$

The goal is to find a global minimum of this function subject to a set of constraints that will be developed in the next sections.

It is important to add that, in Euclidean calibration, the matrix $\mathbf{M}_i$ is the result of the product of two matrices: an upper-triangular matrix of intrinsic parameters $\mathbf{A}_i$ and a rotation matrix. Such matrices can be obtained using RQ - decomposition of $\mathbf{M}_i$ matrix. It is important to notice that the result of such decomposition is unique if $rank(M_i) = 2$. Thus, by computing $\mathbf{M}_i$ as the product of upper-triangular matrix and a rotation matrix, we ensure that the obtained calibration is an Euclidean one.

4.3 Regularization

Many optimization algorithms encounter the following problem: they do not take into account the physical meaning of the optimization parameters, nor the stochastic nature of noise. Thus, due to the presence of noise, they find a solution that actually makes the global error smaller, but the values of parameters do not correspond to a realistic model. Moreover, the autocalibration problem may be considered ill-posed [27] and strongly non-linear due to presence of multiplication of sine and cosine functions. The method allowing to compensate the issue is called regularization and more specifically Tikhonov regularization [28]. The idea behind it consists in introducing more knowledges about the system, by adding weight coefficients to each of the optimized parameter. It allows to improve the likelihood that the minimization converges to the desired solution.

In current cost function (25), all parameters have the same weight. However, we would like to *tell* the algorithm that the values of intrinsic parameters should be close to the initial ones and that the biggest impact should be made on the values of out-of-plane rotations. In other words, during the optimization, the error should be minimized mainly by adapting motion parameters and not the intrinsic ones.

Mathematically, it translates in adding a standard regularization term of the following form:

$$r = \|\Gamma\xi\|_F^2 \quad (26)$$

where Γ is a regularization diagonal square matrix that has the same number of rows as ξ. Each value on the diagonal represents the desirable impact of corresponding parameter, i.e. the weight. The bigger the value, the less the algorithm would *tend* to change the given parameter. Note that the definition of this regularization term is currently based on the assumption of a normal distribution of parameters, but this model will be refined in futur works.

In this work, we defined all weights as powers of 10:

- 100, for intrinsic parameters
- 0.1, for slope angles θ_1 and θ_2
- 0.01, for out-of-plane rotation ρ

These values were found well suited experimentally.

Finally, the cost function with regularization term is written as:

$$f_r(\xi) = \|\mathcal{W}_r - \mathcal{M}(\xi)\mathcal{M}^+(\xi)\mathcal{W}_r\|_F^2 \quad + \quad \|\Gamma\xi\|_F^2 \quad (27)$$

4.4 Bound constraints

Finding a minimum of the objective function obtained previously is the subject of nonlinear optimization, which is typically a challenge without any additional information and thorough understanding of the nature of parameters. In presented formulation of the objective function all of the parameters ξ have an actual geometric meaning which allows to largely reduce the search space of the solution by implementing the bound constraints. It is all the more pertinent for global optimization, where starting points are actually generated inside predefined bounds. It means that tighter bounds lead to faster convergence and higher probability of finding the solution.

The constraints for the elements of $\mathbf{M}_i$ matrix can be defined as follows. First, as regards the matrix of intrinsic parameters, as the common scale factor f is factored out, the constraints can be easily imposed: the value of aspect ratio should be close to one, because the pixels are generally squared, and the skew factor should be close to zero, which would denote that **x** and **y** axis of camera are perpendicular. Furthermore, as the metric calibration can not be obtained (the only possible is the Euclidean "up-to-scale" calibration), in case of constant focal length, the value of f can be fixed to any positive real value, e.g., to one. Secondly, the rotation matrix is decomposed into a sequence of three elemental rotations in a spherical coordinate system with angles θ_1, ρ and θ_2. Hereafter, we speak about intrinsic rotations which means that they don't occur about the axes of the fixed coordinate system, but about the axes of the rotating coordinate system, which changes its orientation after each elemental rotation. It results in a following constraints for the angles: θ_1 and θ_2 can vary in a range of $(-90°; 90°)$. The angle ρ, the out-of-plane rotation may vary in range that depends on the acquisition process. In typical situation, this angle is equal to several degrees. Here it will be fixed in the

Table 1 Examples of initial values and bound constraints for autocalibration. The rotation of the first camera is fixed and equal to $\mathbf{I}_{2\times 3}$.

	Initial value	Bounds
Aspect ratio α	1	$[0.5;1.5]$
Skew s	0	$[-0.5;0.5]$
θ_1	0	$[-90°;90°]$
ρ	0	$[-90°;90°]$
θ_2	0	$[-90°;90°]$
Total number of varying parameters	$2+3(N_{im}-1)$	

range $(-90°;90°)$. Thus, the constraint for all elements of $\mathbf{M}_i$ matrix are combined in Table 1.

Here, constraints values are chosen larger than what is needed, in order to demonstrate the high convergence range of this method. However, applications could be done with reduced bounds if preliminary data are known.

5 Problem solution based on global search

At this point, two cost functions are defined for autocalibration, $f(\xi)$ and $f_r(\xi)$, the latter one containing the regularization term. Generally, local optimization is often used in computer vision and autocalibration in particular, which works well for convex problems. However, for strongly non-linear cost functions, in order to find the minimum, it is preferable to use global optimization algorithms.

Global optimization algorithms can be subdivided in two main classes [29]: deterministic and probabilistic. In case of deterministic algorithms, the search space can be subdivided into multiple pieces, similarly to the divide and conquer strategy. This step is then followed by the exploration of each smaller region by a local solver. After that, the results are combined and the best one is taken as a global minimum. Deterministic methods can provide a certain level of assurance that the global optimum will be located. However, even if they can guarantee that the solution found is the global one, no algorithm can do it in a finite time [30]. It comes from the fact that the smaller the regions on which the space search is subdivided, the more times the local solver should be launched. At the same time, it becomes evident that if the sizes of the regions are close to zero, the likelihood of finding a global minimum increases. In return, the probabilistic algorithms generate the solution based on random variables. They include algorithms like simulated annealing and evolutionary algorithms. In this case, the solution can be found with relatively high time efficiency, however, the global optimality can not be guaranteed. In present work, we decided to apply both types of algorithms: Scatter Search with a local solver, presented by *Ugray et al.* in [31], and the Genetic Algorithm [32]. For both algorithms, their MatLab implementations were used.

The Scatter Search algorithm (further referenced as GS for Global Search) is a population based meta-heuristic algorithm devised to intelligently perform a search on the problem domain algorithm. It consists in a generation of multiple trial points within finite bounds, which are candidate starting points for a local solver. These are then filtered to provide a smaller subset from which the solver attempt to find a local optimum. Then it evaluates remaining possible candidates and start a local solver. The best solution is retained as the global minimum of the problem. More detailed description of the scatter search algorithm is given in [33].

The Genetic Algorithm (GA) is an heuristic algorithm of search, which is used to solve different optimization problems using random search, mutation and variation of parameters. It belongs to the class of evolutionary algorithms, which generate solutions by the techniques inspired by natural evolution. GA differs from other evolutionary algorithms by putting an accent on the use of cross-over operator, which conducts the operation of recombination of solution candidates.

6 Tests on virtual images

6.1 Accuracy on nominal case

In order to find the best solution, we tested four combinations:

- GSR, scatter search optimization with regularization
- GS, scatter search optimization without regularization
- GAR, genetic algorithm optimization with regularization
- GA, genetic algorithm optimization without regularization

As stated previously, MatLab implementations of these algorithm were used. Their performance were compared to the state-of-the-art optimization algorithm of *Long Quan* [13], refereed as LQ in the text. The factorization method presented by Tomasi and Kanade [9], refereed as TK, will be also used as reference. It consists in inferring scene geometry and camera motion from a stream of images, recovering shape and motion under orthography, without computing depth. This method mainly uses the singular value decomposition of the measurement matrix $\hat{\mathcal{W}}$ to find the rotation matrix and the shape matrix. This method stays a standard to analyze the displacements of an object from multiple views.

A sequence of four synthetic images of a diamond (see Fig. 6) was used. The 3D structure and the cameras parameters are known, so the measurement matrix contains perfect noise-free correspondences. The faces between vertices are given only in a purpose of better visualization of the object. As it was mentioned previously, it is not possible to obtain the metric calibration without any additional information on

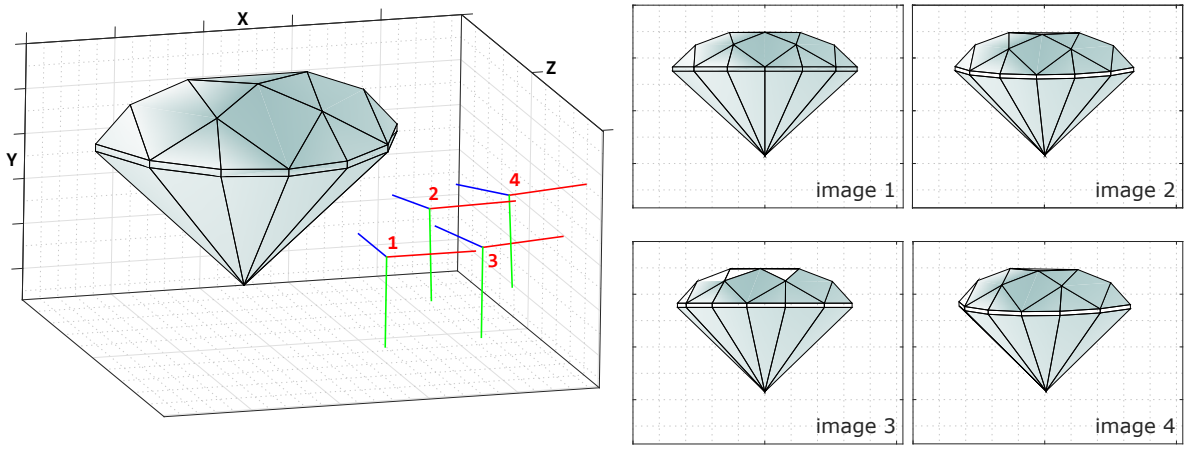


Fig. 6 Four images of diamond sequence with 3D view. Cameras are represented by local coordinate axes.

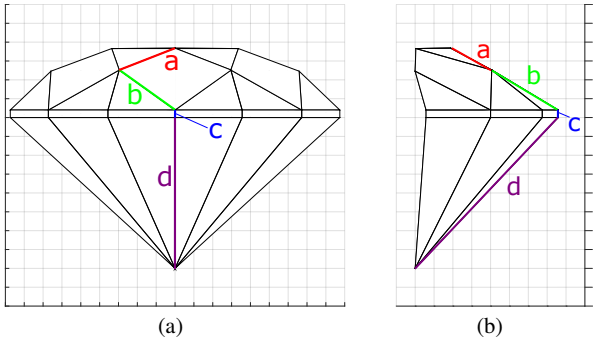


Fig. 7 Front (*a*) and side view (*b*) of the virtual object. The marked sides are used for the analysis of performance of the autocalibration algorithms.

the object structure or camera magnification. Thus, in order to analyze the results, only the properties that are preserved under similarity transformation might be used. Among them are ratio of lengths and angles. For the analysis four vertices of the object were chosen: a, b, c, d (see Fig. 7). The real values of the ratios of lengths and the angles between vertices (α_{ab}, α_{bc}, α_{cd}) as well as the estimated ones are presented in Table 2. It can be noticed that all global methods give a similar results, the estimation error for both ratios and angles is lower than thousandth of percent. This result is not highly significant, but confirm the viability of the presented approach and allows to proceed to more tests on noisy or real images.

It is also interesting to look at the execution time (Table 3). It was measured for MatLab R2017b implementation of global optimization on the computer with the following parameters: Windows10 x64, 3.80 GHz Intel Xeon CPU, 32 GB of RAM. One can remark, that for the same output results, the genetic algorithm takes 100 times more time than scatter search method. Thus, genetic algorithm will no longer be used for tests.

6.2 Robustness to noise

We have several methods (LQ, GS, GSR,TK) that perform equally well for noise free images. However, this situation is far from reality as the set of feature coordinates always contain noisy measurements. Hence, it is important to test the performance of the algorithms at different noise levels. The procedure is exactly the same as above, except for a Gaussian noise with different variance is added to 2D image coordinates. For every noise level, the autocalibration was repeated 100 times. The results are represented in Fig. 8.

At this point we remark that GSR and TK outperforms both the LQ and GS methods. Even for high noise level, the error for estimation of length ratios stays inferior to 2%. In contrast, it may reach 4.5% for LQ method and 3.5% for the scatter search optimization without regularization (GS). The situation with angles is similar, but the difference between methods is lower. These results prove at the same time the efficiency of presented method and the importance of regularization term.

6.3 Convergence range

As it was already shown, global optimization is a time consuming technique. Typically, for a sequence of four images with 22 points, the autocalibration takes about 0.5 seconds. So it is important to verify if it is worth using global optimization instead of local one. To test that, we conducted the following experiment. Once again, four images are obtained in a way that the out-of-plane rotation between the first image and the second one is 5 degrees. Then, we run autocalibration with different initial conditions using GSR method and the same method but with Levenberg-Marquardt (LM) local optimizer (LMR, for Levenberg-Marquardt with regularization). The results are summarized in Table 4.

Table 2 Comparison of performance of different algorithms on a noise free sequence of synthetic images.

	Ratios of lengths			Angles		
	a/b	b/c	c/d	α_{ab}	α_{bc}	α_{cd}
Real value	0.761	12.661	0.036	94.028	114.747	136.544
LQ	0.761	12.661	0.036	94.028	114.747	136.544
GSR	0.761	12.664	0.036	94.030	114.746	136.533
GS	0.761	12.664	0.036	94.021	114.743	136.534
GAR	0.761	12.664	0.036	94.030	114.746	136.533
GA	0.761	12.664	0.036	94.021	114.743	136.534
TK	0.761	12.661	0.036	94.028	114.747	136.544
Errors:						
$\varepsilon_{LQ},\%$	6.56e-13	4.21e-14	1.34e-13	4.53e-13	5.70e-13	6.24e-14
$\varepsilon_{GSR},\%$	5.91e-03	2.97e-02	1.91e-02	2.69e-03	1.71e-03	8.15e-03
$\varepsilon_{GS},\%$	2.75e-03	3.19e-02	1.70e-02	7.07e-03	3.99e-03	7.08e-03
$\varepsilon_{GAR},\%$	5.91e-03	2.97e-02	1.91e-02	2.68e-03	1.71e-03	8.14e-03
$\varepsilon_{GA},\%$	2.75e-03	3.19e-02	1.70e-02	7.07e-03	3.99e-03	7.07e-03
$\varepsilon_{TK},\%$	1.75e-13	4.91e-13	4.48e-13	4.08e-13	3.96e-13	1.25e-13

Table 3 Mean execution time of different autocalibration methods for 4 images and 22 points.

Method	LQ	GSR	GS	GAR	GA	TK
Time, s	0.0156	0.408	0.842	109.118	101.074	0.0011

As expected, with LMR algorithm, optimization suffers from multiple local minima of the cost function, so is needed a very good initial estimate to use this algorithm. In contrast, global optimization allows indeed the arbitrary choice of starting point as it is capable to compensate the error of 45 degrees.

7 Tests on real images

Five image datasets were used:

- affine camera images:
  - **hotel**: 10 images from the hotel image sequence [34].
- SEM images from [35]:
  - **brassica**: 4 images of pollen grain of white turnip plant from a Hitachi S-4800 FE-SEM.
  - **grid**: 5 images of TEM copper grid from a Hitachi S-4800 FE-SEM.
- SEM images acquired at FEMTO-ST Institute, taken using a SEM Zeiss AURIGA 60:
  - **pot**: 7 images of pollen grain of aquatic, mostly freshwater, plant of the family *Potamogetonaceae*.
  - **pot2**: 6 images of another *Potamogetonaceae* pollen grain.

The **hotel** sequence is the image dataset used by *Long Quan* to validate his algorithm. In our implementation, 196 matches were extracted and the results of autocalibration, as expected, were exactly the same for all four methods (TK, LQ, GS, GSR). An example of camera poses as well as the sparse 3D calibration is shown in Fig. 9.

Back to SEM images, three datasets were analyzed. As all image datasets were taken using SEM, there is no available information on the location of the camera. Thus, we will use two criteria allowing to judge on the algorithm performance. First, the respect of metric constraints and, secondly, the reprojection error (Fig. 10). It is important to note that we use a leave-one-out method to do the reprojection: for a set of N images, we extract each image one time to estimate the 3D-points, then reproject these points according to the estimated pose for this image. In this way, the method's robustness has to be better than if the 3D-points are defined from the N images.

We consider that metric constraints are respected if the aspect ratio is within the interval $(0.9; 1.1)$ although we know that it has to be close to one. For LQ and GS method, the constraints are never respected. The TK cannot estimate the metric constraints. GSR algorithm gives the desired result for all datasets as the constraints on aspect ratio and skew are already included in cost function formulation.

In terms of reprojection error, the algorithms gave good results for TK and GSR. This result is important because it proves that even if we prevent the change of intrinsic parameters, the GSR algorithm succeeds in keeping the same level of accuracy as TK.

Obviously, these two criteria do not allow full validation of the autocalibration: one can consider them as mandatory but not sufficient. Unfortunately, when working at such scales and with small objects, there is no measurement device capable to provide the ground truth. However, for some datasets, we know the actual angle of tilt: 3 degrees for **brassica**, 7 degrees for **grid**, and 3 degrees for **pot**. For other datasets, the rotation between images were performed as a combination of two articular robot movement which was not calibrated, which means that the angle of out-of-plane rotation cannot be extracted.

Table 4 Comparison of convergence range between local optimizer (Levenberg-Marquardt, LMR) and global optimizer (GSR). Tests were run on noise-free images.

Initial ρ_{12}, in degrees	-20	0	1	2	3	4	5	6	7	8	20	45
LMR	5.23	0.00	1.69	4.07	4.67	4.81	5.00	5.06	5.08	5.10	5.23	5.4
GSR	5.00	5.00	5.00	5.00	5.00	5.00	5.00	5.00	5.00	5.00	5.00	5.00

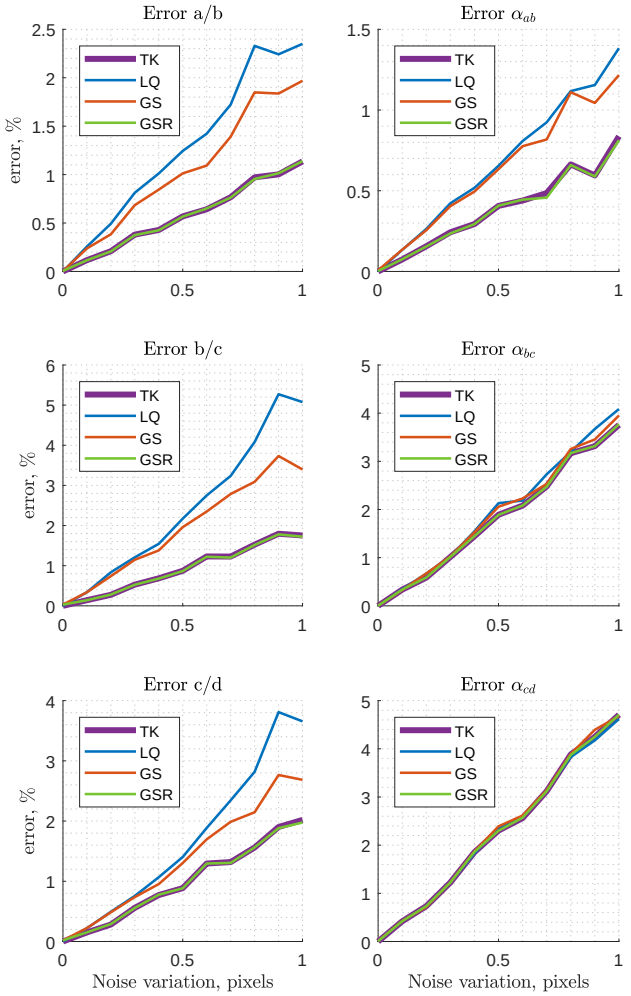


Fig. 8 Robustness of different autocalibration algorithms to noisy measurements. TK and GSR obtain best results, mainly overlaid on curves.

To compare the rotation angles, the following procedure was performed. Once the sequence passed through autocalibration, the rotation matrices for every frame were extracted and then transformed into axis-angle representation. Therefore, if the rotation were performed in one movement, these angles should match the angles given by robot sensor. The results are presented in Table 5. The difference do not exceed 0.21 degrees on all tries.

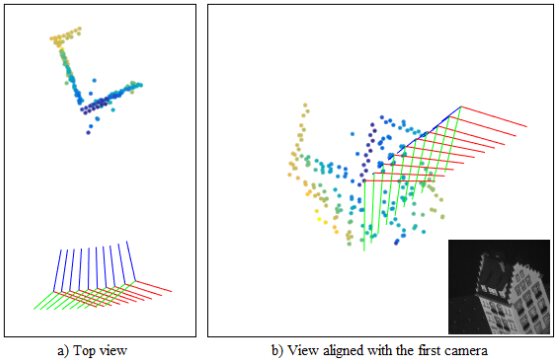


Fig. 9 Result of autocalibration for **hotel** image dataset.

Table 5 Comparison between rotation angles in degrees given by robot sensors (true), and autocalibration algorithm (estimated).

	brassica		**grid**		**pot**	
	true	estimated	true	estimated	true	estimated
ρ_{12}	3	2.9425	7	6.9103	3	3.0312
ρ_{23}	3	3.0182	7	7.0851	3	3.0965
ρ_{34}	3	2.8270	7	6.7993	3	3.0723
ρ_{45}	-	-	7	6.8596	3	3.1131
ρ_{56}	-	-	7	-	3	2.9628
ρ_{67}	-	-	-	-	3	3.0676

8 Conclusion

In this paper, a new method of autocalibration for affine camera was presented. All its components have been computed: model as well as motion parameters. The presented method, being based on global optimization, has the following advantages comparing to the state-of-the-art techniques:

- All metric constraints are imposed directly on the optimized parameters. Moreover, with regularization, it is possible to guide the optimization towards the desired result, i.e. add a penalty score for excessive change of such parameters as aspect ratio and skew.
- All optimization parameters have a physical meaning. It means that all known elements about camera location can be easily imposed. For example, one can impose the equality of out-of-plane rotation if the angle of tilt is known to be constant for all image sequence.
- Thanks to the deep analysis of camera properties, the final number of optimized parameters is low and good initial estimate is provided from the elements of funda-

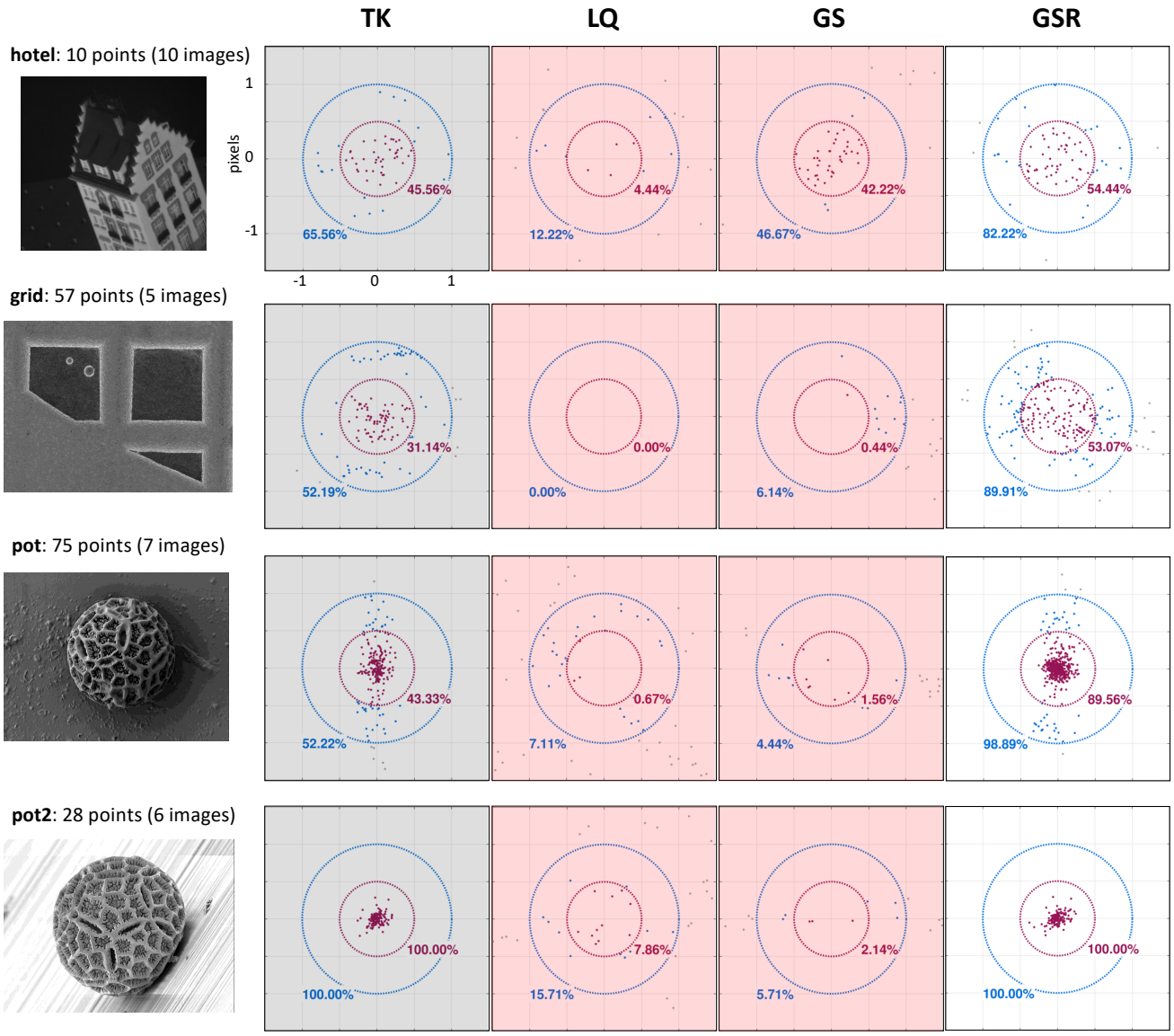


Fig. 10 Analysis of reprojection error for different autocalibration algorithms: TK, LQ, GS, GSR. Red background means that metric constraints are not respected: typically, aspect ratio is greater than 1.1 or lower than 0.9. Grey background means that metric constraints are not calculated. For points inside circles, the reprojection error is lower than: 1 pixel for blue circle, 0.5 pixel for red circle. Numbers reflect the percentage of points inside the corresponding circle.

mental matrices. These are the key success factors of optimization. Moreover, in contrast to classical bundle adjustment techniques, the 3D points are excluded from optimization process.

– The fact of using global optimization ensures high convergence range.
– The execution time is relatively low: less than 0.5 seconds for 4 images. It can be improved, first, by implementing global optimization in C++. Secondly, the optimization for multiple starting points may be carried out in parallel which would also decrease the optimization time.

With regard to microscopy community, we presented the first autocalibration algorithm for Scanning Electron Microscope.

It is worth adding that the crucial step in all autocalibration algorithms is the feature matching across the images. In case of affine camera less points are needed to constraint camera parameters, because of the reduced number of degrees of freedom. As it was shown before, we used the number of features from 20 to 200. If it is possible to increase the number of matches without loss of their quality, it will improve the final results, however, will slightly increase the execution time. Though, the quality of the correspondences

is much more important than their quantity: even if a feature is an inlier, its quality may vary. Outliers are rejected at the step of fundamental matrix estimation.

References

1. Joseph L. Mundy and Andrew Zisserman. *Geometric invariance in computer vision*, volume 92. MIT press Cambridge, 1992.
2. Guanghui Wang and Jonathan Wu. *Guide to Three Dimensional Structure and Motion Factorization*. Springer London, London, 2011.
3. Zhengyou Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *IEEE Int. Conf. Computer Vision (ICCV'99)*, volume 1, pages 666–673, 1999.
4. Olivier D Faugeras, Q-T Luong, and Stephen J Maybank. Camera self-calibration: Theory and experiments. In *European Conf. Computer Vision (ECCV'92)*, pages 321–334, 1992.
5. Richard I Hartley. Projective reconstruction and invariants from multiple images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16:1036–1041, 1994.
6. Peter Sturm and Bill Triggs. A factorization based algorithm for multi-image projective structure and motion. In *European Conf. Computer Vision (ECCV'96)*, pages 709–720. 1996.
7. Yuchao Dai, Hongdong Li, and Mingyi He. Element-wise factorization for n-view projective reconstruction. In *European Conf. Computer Vision (ECCV'10)*, pages 396–409. 2010.
8. John Oliensis. A critique of structure-from-motion algorithms. *Computer Vision and Image Understanding*, 80:172–214, 2000.
9. Carlo Tomasi and Takeo Kanade. Shape and motion from image streams under orthography: a factorization method. *Int. J. Computer Vision*, 9:137–154, 1992.
10. Marc Pollefeys and Luc Van Gool. Stratified self-calibration with the modulus constraint. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21:707–724, 1999.
11. Bill Triggs. Autocalibration from planar scenes. In *European Conf. Computer Vision (ECCV'98)*, pages 89–105. 1998.
12. Peter Sturm. Critical motion sequences for monocular self-calibration and uncalibrated euclidean reconstruction. In *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR'97)*, pages 1100–1105, 1997.
13. Long Quan. Self-calibration of an affine camera from multiple views. *Int. J. Computer Vision*, 19:93–105, 1996.
14. Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.
15. Andrea Fusiello, Arrigo Benedetti, Michela Farenzena, and Alessandro Busti. Globally convergent autocalibration using interval analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26:1633–1638, 2004.
16. Manmohan Chandraker, Sameer Agarwal, Fredrik Kahl, David Nistér, and David Kriegman. Autocalibration via rank-constrained estimation of the absolute quadric. In *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR'07)*, pages 1–8, 2007.
17. Stuart B. Heinrich, Wesley E. Snyder, and Jan-Michael Frahm. Maximum likelihood autocalibration. *Image and Vision Computing*, 29:653–665, 2011.
18. Severin Amelinckx, Dirk Van Dyck, J Van Landuyt, and Gustaaf Van Tendeloo. *Electron microscopy: principles and fundamentals*. John Wiley & Sons, 2008.
19. Joseph I Goldstein, Dale E Newbury, Joseph R Michael, Nicholas WM Ritchie, John Henry J Scott, and David C Joy. *Scanning electron microscopy and X-ray microanalysis*. Springer, 2017.
20. Eraldo Ribeiro and Mubarak Shah. Computer vision for nanoscale imaging. *Machine Vision and Applications*, 17(3):147–162, 2006.
21. Joseph Goldstein, Dale E. Newbury, David C. Joy, Charles E. Lyman, Patrick Echlin, Eric Lifshin, Linda Sawyer, and J.R. Michael. *Scanning electron microscopy and x-ray microanalysis*. Springer, 2003.
22. Nicolas Cornille, Dorian Garcia, Michael A. Sutton, S. McNeill, and Jean-José Orteu. Automated 3-d reconstruction using a scanning electron microscope. In *SEM Conf. Exp. and Appl. Mech.*, 2003.
23. O. Sinram, M. Ritter, S. Kleindick, A. Schertel, H. Hohenberg, and J. Albertz. Calibration of a sem, using a nanopositioning tilting table and a microscopic calibration pyramid. *ISPRS Archives*, 34:210–215, 2002.
24. Le Cui and Eric Marchand. Calibration of scanning electron microscope using a multi-image non-linear minimization process. In *IEEE Int. Conf. Robot. Autom. (ICRA'14)*, pages 5191–5196, 2014.
25. Andrey V. Kudryavtsev, Sounkalo Dembélé, and Nadine Piat. Full 3d rotation estimation in scanning electron microscope. In *Int. Conf. on Intelligent Robots and Systems (IROS'17)*, pages 42–47, 2017.
26. J. He, R. Zhou, and Z. Hong. Modified fast climbing search autofocus algorithm with adaptive step size searching technique for digital camera. *IEEE Trans. Consum. Electron.*, 12:244–252, 2003.
27. Jacques Hadamard. Sur les problèmes aux dérivées partielles et leur signification physique. *Princeton University Bulletin*, pages 49–52, 1902.
28. Andrey Nikolaevich Tikhonov, Vasiliy Yakovlevich Arsenin, and Fritz John. *Solutions of ill-posed problems*, volume 14. Winston Washington, DC, 1977.
29. Thomas Weise. Global optimization algorithms - theory and application. *Self-Published,*, 2009.
30. Carmen G Moles, Pedro Mendes, and Julio R Banga. Parameter estimation in biochemical pathways: a comparison of global optimization methods. *Genome Research*, 13:2467–2474, 2003.
31. Zsolt Ugray, Leon Lasdon, John Plummer, Fred Glover, James Kelly, and Rafael Martí. Scatter search and local NLP solvers: A multistart framework for global optimization. *INFORMS J. Comput.*, 19:328–340, 2007.
32. David E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Publishing Company, 1989.
33. Fred Glover. A template for scatter search and path relinking. *Lecture notes in computer science*, 1363:13–54, 1998.
34. Conrad J Poelman and Takeo Kanade. A paraperspective factorization method for shape and motion recovery. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19:206–218, 1997.
35. Ahmad P Tafti, Andrew B Kirkpatrick, Jessica D Holz, Heather A Owen, and Zeyun Yu. 3dsem: A dataset for 3d sem surface reconstruction, 2015.