# An Iterative Solution for the Structure from Motion Problem Resolution 

A. El-Attar, H.Tairi, M. Karim<br>Faculty of Sciences Dhar EL Mahraz<br>University of Sidi Mohammed Ben Abdellah - Fez. Algeria<br>adnane1230@gmail.com, \{htairi, karim_lessi\}@yahoo.fr


#### Abstract

This paper presents a robust algorithm for the Structure From Motion (SFM) problem resolution using a step-by-step non-linear optimization scheme. The proposed algorithm is composed by two main parts. In the first part a projective reconstruction is computed. Here, the reconstruction is initialized based on the epipolar geometry of the scene and then refined using a robust non-linear optimization. To upgrade the reconstruction to a metric one, the second part of the algorithm consists on computing the camera intrinsic parameters. To deal with the drawback of computing all intrinsic parameters at once and avoid the possibility of converging to a local minimum, the matrix of camera intrinsic parameters is firstly initialized and then each parameter is estimated alone in a multi-stage scheme. Once computed, the intrinsic parameters are introduced in a state-of-the-art Multi-View Stereo (MVS) to generate the dense 3D model of the scene. MVS methods take images with camera internal parameters as input and produce dense 3D models with accuracy nearly on par with laser scanners.


Keywords: Structure From Motion, 3D reconstruction, Camera Auto-Calibration, Projective Reconstruction, Multi-stage Algorithm
Received: 30 May 2012, Revised 9 July 2012, Accepted 17 July 2012
© 2012 DLINE. All rights reserved

## 1. Introduction

During the last two decades, the Structure From Motion (SFM), from active sensors, has been a very active research topic in computer vision. Starting by the acquisition of a set of uncalibrated images of a fixed scene by a moving camera (or vice-versa, i.e. moving object and fixed camera); the goal of SFM is to recover the 3D reconstruction of the scene in the Euclidean space. The general scheme of an SFM algorithm may be described as follow: First, feature descriptors of the different images are detected and matched, across all views, using a robust optimization method, such as RANSAC [8], this leads to a set of accurate and robust matches $p(x, y) \Leftrightarrow p^{\prime}\left(x^{\prime}, y^{\prime}\right)$. Second, a projective reconstruction is retrieved based on the computed matches. Projective reconstruction means that the computed 3D model is similar to the real objects in the scene up a transformation matrix $H$ called homography, the role of $H$ is to preserve some geometrical properties of the scene such as orthogonality, parallelism and distances ratios (figure 1). Third, the upgrade of the projective reconstruction to a metric one; this step requires the knowledge of the camera intrinsic parameters, which allow the computation of the homography. To this end, a very important step of camera auto-calibration is accomplished in this stage to compute the camera internal parameters.

The accuracy and robustness of the projective reconstruction step depend highly on the accuracy of the estimation of the scaled data measurement matrix $W$. Nowadays; algorithms that aim to better estimate $W$, in the projective space, are more and more exposed. Several researchers [3, 4, 5, 6] have recently proposed some extensions of the Sturm / Triggs (S / T) algorithm [1]. Most of them are based on an initialization of the projective depth $\lambda$ to 1 , then an extraction of the projective structure $X$, use this structure to refine ë and use the new $\lambda$ to refine the Structure and the projection matrices for each image $P_{i}$.

As stated in [2]; the weakness of previous methods is the fact that: $1^{\text {st }}$ the use of $\lambda$ equal to 1 is not suitable because in this case the algorithm may converge to a local minimum. $2^{\text {nd }}$ These methods have not been theoretically proved. $3^{\text {rd }} A$ certain lack of the theoretical support for some basic notions that they should develop.

We propose an algorithm for the estimation of both $\lambda$ and in an iterative way to avoid the usage of $\lambda=1$ and thus better retrieve the projective reconstruction $\left(P_{i}, X_{j}\right)$.


Figure 1. Different kind of reconstructions of the scene (see section 5. A for more details about the different transformations)

On the other hand; many successful results had been obtained, for camera auto-calibration, based on the rigidity of the scene. In the literature, three main approaches are distinguished: first, those based on the Kruppa's equations [9, 11, 13], they require the epipolar geometry of each pair of views and consist of two independent equations in the elements of the image of the absolute conic. Second, methods making a direct calculation of the homography $H$ to rectify the reconstruction from "projective-to-metric" [11, 13, 14, 16]. Triggs et al [12] introduced an algorithm based on the absolute dual quadric where embedded the plane at infinity and the intrinsic parameters of the camera in a compact way. This model has been used in [15] to introduce a nonlinear and linear algorithms to deal with the case of varying intrinsic parameters. Third, algorithms that use a stratified approach, based on the affine rectification of a projective reconstruction, and find linearly a transformation "Affine-Metric", this has the advantage to deal with the drawback of calculating many parameters at once when upgrading directly the projective calibration to a metric one. The first step for this kind of algorithms is accomplished by an exhaustive search for the plane at infinity [10], the subsequent determination of the calibration matrix is then relatively simple because there exist a linear solution.

In second part of this paper we introduce an algorithm to deal with the disadvantage of computing several independent parameters at once using the Absolute Dual Quadric-based method described above. Therefore, we firstly initialize the camera intrinsic parameters, then compute each parameter alone in a multi-stage scheme and finally refine all of them at once, in a nonlinear iterative way, to ensure getting accurate results.

## 2. Paper Outline

This paper is divided into two main parts; the 1st one introduces an algorithm concerning the extraction of the projective reconstruction while the 2nd part represents the method to compute the camera intrinsic parameters and thereafter the derivation of the 3D reconstruction of the scene. A previous version of the two algorithms had been published in [17] and [18]. The differences here are, on one hand, the acceleration of the 1st part with the advantage of getting stable and robust results. On the other hand the simplification of the camera auto-calibration phase and the usage of the computed parameters to upgrade the reconstruction from projective to metric. Finally the dense 3D model of the scene is extracted based on the computed parameters
142 Journal of Information \& Systems Management Volume 2 Number 3 September 2012
using a Multi-View Stereo method (MVS) such as [20, 21].
The rest of the paper is organized as follows: section 3 presents some preliminaries and notations used in the paper. Section 4 introduces the method to compute the projective reconstruction, and then the auto-calibration stage and 3D model extraction is described in section 5 . Experiments are in section 6 and finally section 7 concludes the paper.

## 3. Preliminaries

### 3.1 Notations

In this paper we will use the following notations: an entity $L$ in the Euclidean space will be noted with a subscript $E: L_{E}$, a 3D world points X will be denoted by a homogeneous 4 -vector ( $\mathrm{X}, \mathrm{Y}, \mathrm{Z}, 1$ ) and a 2 D image points x by homogeneous 3 -vector ( $\mathrm{x}, \mathrm{y}$,
1). Suppose a camera observing a scene composed of 3 D points $X_{\mathrm{j}}$, the perspective projection of the scene points into the image plane is given by figure 2 .

Mathematically, we express this projection as follow:

$$
\begin{equation*}
\lambda_{\mathrm{ij}} \mathrm{X}_{\mathrm{j}}=\mathrm{P}_{\mathrm{j}} \mathrm{X}_{\mathrm{j}} \tag{1}
\end{equation*}
$$

For $\mathrm{i}=1, \ldots, \mathrm{~N}_{\mathrm{I}}$ and $\mathrm{j}=1, \ldots, \mathrm{~N}_{\mathrm{p}}$
Where

$$
\begin{equation*}
\mathrm{P}_{\mathrm{i}}=\mathrm{K}_{\mathrm{i}}\left[\mathrm{R}_{\mathrm{i}} \mathrm{t}_{\mathrm{i}}\right] \tag{2}
\end{equation*}
$$

$P_{i}$ is the camera projection matrix. $\mathrm{R}_{\mathrm{i}}$ and $\mathrm{t}_{\mathrm{i}}$ are the rotation matrix and the translation vector respectively. $\mathrm{K}_{\mathrm{i}}$ is the calibration matrix and is expressed as follow:

$$
\mathrm{K}_{\mathrm{i}}=\left(\begin{array}{ccc}
\mathrm{f}_{\mathrm{i}} & \mathrm{~s}_{\mathrm{i}} & u_{0 \mathrm{i}}  \tag{3}\\
0 & \mathrm{k}_{\mathrm{i}} \mathrm{f}_{\mathrm{i}} & \mathrm{v}_{0 \mathrm{i}} \\
0 & 0 & 1
\end{array}\right)
$$

Where $f_{i}$ is the focal length; $k_{i}$ is the aspect ratio; $s_{i}$ is the skew and $\left(u_{0 i}, v_{0 i}\right)$ are the coordinate of the principal point.
For $\mathrm{N}_{\mathrm{I}}$ image and $\mathrm{N}_{\mathrm{p}} 3$ D points, equation (1) can be written using its factorized matrix form:

Equivalent to: $W=P_{3 N_{I} \times N_{P}} \times \chi_{4 \times N_{P}}$
Where $W_{3 N_{l} \times N_{P}}$ is the Scaled Data Measurement Matrix, it's composed by the image points $\mathrm{x}_{\mathrm{ij}}$ multiplied by a non-zero scale factor called the projective depth: $\lambda_{\mathrm{ij}} \in \mathbb{R}^{N_{*} * N_{P}} \cdot P_{3 N_{I} \times 4}$ is the Motion Matrix, composed by the camera projection matrices $\mathrm{P}_{\mathrm{i}}$ in each view, and $\chi_{4 \times N_{P}}$ is the Structure Matrix composed by the 3 D points $X_{i}$ expressed in their homogenous form $X_{j}=(X, Y, Z, 1) \in \mathbb{R}^{4}$.


Figure 2. The Perspective Projection of the Scene into the image plane

## 4. Computation of the projective reconstruction

Let $\mathrm{F}_{12}$ be the fundamental matrix between the first and the second view, the projection matrices of these views can be expressed as follow [6]:

$$
\left\{\begin{array}{c}
P_{1}=(I \mid 0)  \tag{5}\\
P_{2}=\left(\left(e_{21}\right) \times F_{12}+e_{21} v^{T} \mid \alpha e_{21}\right)
\end{array}\right.
$$

Where $\left(e_{21}\right)_{x}$ is the antisymmetric matrix corresponding to the epipole in the second view. v is an arbitrary ( $3 \times 1$ ) vector and $\alpha$ is a non-zero arbitrary scalar. Once $\mathrm{P}_{\mathrm{i}}$ are obtained, one may retrieve the projective structure by triangulation [7, 19]. However, this is computationally expensive because it consists on the minimization of the reprojection error of 3 D points $\mathrm{X}_{\mathrm{j}}$ :

$$
\begin{equation*}
\min \left(\operatorname{dist}(x, P X)^{2}+\operatorname{dist}\left(x^{\prime}, \mathrm{P}^{\prime} \mathrm{X}\right)^{2}\right. \tag{6}
\end{equation*}
$$

Where dist $(a, b)$ is the Euclidean distance between $a$ and $b$.
Instead of minimizing (6) we prefer to use a simple method to first correct feature matches in order to have only correct matches $(\mathrm{x}, \mathrm{y}) \Leftrightarrow\left(\hat{\mathrm{x}}^{\prime}, \hat{\mathrm{y}}^{\prime}\right)$. Then we use the SVD decomposition to extract Xj from the corrected matches.

The first order correction of the measured points $x_{i j}=\left(x, y, x^{\prime}, y^{\prime}\right)$ is given by:

$$
\begin{equation*}
\delta_{x}=-J T(J J T)^{-1} \varepsilon \tag{7}
\end{equation*}
$$

And the corrected point is: $\hat{x}=x+\delta_{x}=x-J^{T}\left(J J^{T}\right)^{-1} \varepsilon$
In case of a matching defined by: $\hat{x}^{T^{\prime}} \hat{F x}=0$ the error is: $\varepsilon=x^{T^{\prime}} F x$ and the jacobian is:

$$
\begin{equation*}
J=\frac{\partial \varepsilon}{\partial x}=x\left[\left(F^{T} x^{\prime}\right)_{1},\left(F^{T} x^{\prime}\right)_{2},\left(F^{T} x\right)_{1},\left(F^{T} x\right)_{2}\right] \tag{8}
\end{equation*}
$$

Where $\left(F^{T} x^{\prime}\right)_{1}=f_{11} x^{\prime}+f_{21} y^{\prime}+f_{31}$, and so on for the other element of $J . f_{i j}$ are the elements of the fundamental matrix $F$ with $\mathrm{i}=$ $1,2,3$ and $\mathrm{j}=1,2,3$. In this case, the first order correction is:

$$
\left(\begin{array}{c}
\hat{x}  \tag{9}\\
\hat{y} \\
\hat{x}^{\prime} \\
\hat{x}^{\prime}
\end{array}\right)=\left(\begin{array}{l}
x \\
y \\
x^{\prime} \\
x^{\prime}
\end{array}\right)-\frac{x^{T^{\prime}} F x}{B}\left(\begin{array}{c}
\left(F^{T} x^{\prime}\right)_{1} \\
\left(F^{T} x^{\prime}\right)_{2} \\
(F x)_{1} \\
(F x)_{2}
\end{array}\right)
$$

Where:

$$
\begin{equation*}
B=(F x)_{1}^{2}+(F x)_{2}^{2}+\left(F^{T} x^{\prime}\right)_{1}^{2}+\left(F^{T} x^{\prime}\right)_{2}^{2} \tag{10}
\end{equation*}
$$

Let $\mathrm{p}^{1-3}$ and $\mathrm{p}^{\prime 1-3}$ be the three rows of P and $\mathrm{P}^{\prime}$ respectively. One can observe that:

$$
\left\{\begin{array}{l}
x p^{3 T} X-p^{1} x=0  \tag{11}\\
y p^{3 T} X-p^{2} x=0 \\
x p^{2 T} X-y p^{1} X=0
\end{array}\right.
$$

This can be written as:

$$
A=\left[\begin{array}{l}
x p^{3 T}-p^{1 T}  \tag{12}\\
y p^{3 T}-p^{1 T} \\
x^{\prime} p^{\prime 3 T}-p^{\prime 1 T} \\
y^{\prime} p^{\prime 3 T}-p^{\prime 2 T}
\end{array}\right]
$$

It's clear now that once A is obtained, X can be computed linearly.

Once getting X and P , the projective reconstruction $\left(P_{i}, X_{j}\right)$ can be optimized by minimizing the following criterion:

$$
\begin{equation*}
G=\frac{1}{2} \min \lambda_{i j}\left\|\hat{\lambda}_{b J} x_{i j}-P_{i} X_{j}\right\|_{F}^{2} \tag{13}
\end{equation*}
$$

The solution $J_{\text {init }}$ is used as the initial guess for an iterative optimization procedure to best estimate ë_ij. More concretely, the procedure is completed as follows: Given the data measurement matrix W and its singular value decomposition (SVD): $\mathrm{W}=$ $\mathrm{UDV}^{\wedge} \mathrm{T}$, we define the approximate matrix $\mathrm{W}^{\wedge}$ such as: rankd $\leq 4$ as follow:

$$
\mathrm{W}^{\wedge}=\mathrm{UD}^{\wedge} \mathrm{V}^{\wedge} \mathrm{T}
$$

Where $\widehat{D}^{\wedge}$ is obtained from $D$ by zeroing all its elements except the first four. In other words, $W^{\wedge}$ gives the best approximation among all matrices W whose $\operatorname{rank}(\mathrm{W}) \leq 4$ :

$$
\|\mathrm{W}-\widehat{\mathrm{W}}\|=\min _{\operatorname{range}(\mathrm{Y}) \leq 4}\|\mathrm{~W}-\mathrm{Y}\|
$$

Y is the set of approximate matrices W whose $\operatorname{rank}(\mathrm{Y}) \leq 4$

## 5. Camera Auto-Calibration and Metric Reconstruction:

### 5.1 Problem formulation

Equation (1) can be written in the Euclidean space as follows:

$$
\begin{equation*}
\mathrm{x}_{\mathrm{ij}} \cong \mathrm{P}_{\mathrm{Ei}} \mathrm{X}_{\mathrm{Ej}} \tag{14}
\end{equation*}
$$

Where $\cong$ means that the equality holds up to a scale. From (1) and (14) one can see that the projective reconstruction is related to the Euclidean one by an arbitrary $4 \times 4$ homography:

$$
\left\{\begin{array}{l}
P_{i}=P_{E I} H^{-1}, i=1, \ldots, m  \tag{15}\\
X_{j}=H^{x-1} X_{E j}, j=1, \ldots, n
\end{array}\right.
$$

Therefore, our problem can be stated as follow: Given the projective reconstruction ( $\mathrm{P}_{\mathrm{i}}, \mathrm{X}_{\mathrm{j}}$ ), computed in the previous section, the purpose of auto-calibration is to estimate the best homography $H$ that upgrades the projective reconstruction to a metric one. This is done through the determination of the camera intrinsic parameters, once they are computed, the homography H can be computed linearly.

### 5.2 The Absolute Dual Quadric:

The Absolute Dual Quadric $Q^{*}$ is a degenerate dual quadric represented mathematically by a $4 \times 4$ rank 3 homogenous matrix, its importance for camera self-calibration comes from the fact that it encodes both the plane et infinity $\pi_{\infty}$ and the absolute conic $\omega$ :

$$
\begin{equation*}
\omega_{i}^{*}=K_{i} K_{i}^{T} \cong P_{i} Q_{\infty}^{*} P_{i}^{T} \tag{16}
\end{equation*}
$$

$Q_{\infty}^{*}$ is the Absolute Quadric and $\omega_{i}^{*}$ is the Dual Image Absolute Conic (DIAC) [7]
It follows that $Q_{\infty}^{*}$ projects to the dual image absolute conic:

$$
\begin{equation*}
\omega_{i}^{*}=K_{i} K_{i}^{T} \tag{17}
\end{equation*}
$$

Given (16), constraints on $\omega_{i}^{*}=K_{i} K^{T}$ can be translated into constraints on $\mathrm{Q}_{\infty}^{*}$ using the projection matrices computed previously, thus can be computed in the projective reconstruction using constraints on $\mathrm{K}_{\mathrm{i}}$. We recommend the reference [7] (section 19.3 $\mathrm{p}-462$ ) for a thorough treatment of this operation.

Equation (16) can be used to retrieve metric measurements from a given projective reconstruction, however $\mathrm{Q}_{\infty}^{*}$ should be parameterized in a manner to enforce the constraints on $\mathrm{K}_{\mathrm{i}}$, an easy way to do this is by using a minimum parameterization of $\mathrm{Q}_{\infty}^{*}$, i.e. we put $\left(\mathrm{Q}_{\infty}^{*}\right)_{33}=1$ and compute $\left(\mathrm{Q}_{\infty}^{*}\right)_{44}$ using the rank 3 constraint.

$$
\mathrm{Q}_{\infty}^{*}=\left(\begin{array}{cc}
K K^{T} & -K K^{T} p  \tag{18}\\
-p^{T} K K^{T} & p^{T} K K^{T} p
\end{array}\right)
$$

Where p defines the position of the plane at infinity $\pi_{\infty}=\left(p^{\prime} 1\right)^{\prime}$

Using the above formulation of $\mathrm{Q}_{\infty}^{*}$, camera intrinsic parameters can be extracted by minimizing the following criterion:

$$
\begin{equation*}
\min _{K_{i}} \sum_{\mathrm{i}=1}^{\mathrm{n}}\left\|\frac{\mathrm{~K}_{\mathrm{i}} \mathrm{~K}_{\mathrm{i}}^{\mathrm{T}}}{\left\|\mathrm{~K}_{\mathrm{i}} \mathrm{~K}_{\mathrm{i}}\right\|_{\mathrm{F}}}-\frac{\mathrm{P}_{\mathrm{i}} \mathrm{Q}_{\infty}^{*} \mathrm{P}_{\mathrm{i}}^{\mathrm{T}}}{\left\|\mathrm{P}_{\mathrm{i}} \mathrm{Q}_{\infty}^{*} \mathrm{P}_{\mathrm{i}}^{\mathrm{T}}\right\|_{\mathrm{F}}}\right\|_{\mathrm{F}}^{2} \tag{19}
\end{equation*}
$$

Both elements in (19) should be normalized to eliminate the scale factor in equation (16)

### 5.3 The Multi-stage algorithm

Equation (19) is a nonlinear least square that requires an important stage of initialization, to do this, we will choose the first image center to be centered with the world coordinate frame, in this case we have: $\mathrm{P}_{1}=(\mathrm{I} \mid 0)$. Using this formulation the equation for the first view will be perfectly satisfied, unfortunately, the noise has to be spread over all frames of the sequence, and for this reason we propose to use this parameterization as an initial guess for a multistage algorithm to estimate the intrinsic parameters one by one:

The proposed auto-calibration algorithm is a multi-stage one; at each stage an internal parameter of the calibration matrix K is estimated and the output calibration matrix is used as an initial guess for the estimation of the next parameters. We will start by an initial estimation of the focal length, then compute the other parameters one by one and finally refine all of them at once. To start this, equation (13) will be minimized using the following approximations: assuming a camera with zero skew and the principal point in the image center:

$$
s=0 \text { and }\left(u_{0}, v_{0}\right)=\left(u_{c}, v_{c}\right)
$$

We will also assume a unit aspect ratio, thus we still have only one unknown parameter, i.e. the focal length.
The subsequent of the algorithm is summarized in table 5.1
The complexity of the algorithm in table 6.1 is $\mathrm{O}\left(\mathrm{N}_{\mathrm{I}}\right)$; where $\mathrm{N}_{\mathrm{I}}$ is the number of input images.
Once the intrinsic camera parameters are obtained, the pose of the camera (Rotation + Translation) and the $4 \times 4$ homography that upgrade the reconstruction from projective to metric (Euclidean up to a scale) can be easily computed using a classical linear algorithm.

Later on, the computed entities are used in a MVS algorithm to generate the 3D dense model of the scene. MVS methods take images with intrinsic parameters and pose estimation as input to produce dense 3D models with high accuracy.

### 4.4 Extraction of 3D dense model

MVS algorithms are based on the idea of correlating measurements from several images at once to derive 3D surface information. In the following, we present a brief description of the used MVS algorithm presented in [21] and used to derive the 3D reconstruction based on the computed camera parameters. Here an overlapping view clustering is designed. The goal of view clustering is to find (an unknown number of) overlapping image clusters $\left\{\mathrm{C}_{\mathrm{k}}\right\}$ such that each cluster is of manageable size and each 3D point can be accurately reconstructed by at least one of the clusters. Figure5.1 provides an overview of the algorithm.

The designed algorithm aims to satisfy the following three constraints:
1- Compactness: redundant images are excluded from the clusters
2- Size: each cluster is small enough for an MVS reconstruction.
3- Coverage: MVS reconstructions from these clusters result in minimal loss of content and detail compared to that obtainable by processing the full image set.

More specifically: the purpose is to optimize the set of images $\Sigma_{\mathrm{k}} \mid \mathrm{C}_{\mathrm{k}}$ in the output cluster into:

- Upper bound on the size of each cluster so that an MVS algorithm can be used for each cluster independently: $\forall \mathrm{k},\left|\mathrm{C}_{\mathrm{k}}\right|<\alpha$. Where $\alpha$ is determined by computational resources.
- Enforce the coverage of the 3D reconstruction as follow: a 3D point $P$ is covered if it's sufficiently well reconstructed by the cameras in at least one cluster $\mathrm{C}_{\mathrm{k}}$. More concretely, in order to measure the accuracy achieved at a 3D location by a set of images $C$, a function $f(P, C)$ is defined.
146 Journal of Information \& Systems Management Volume 2 Number 3 September 2012

Accordingly, a point is covered if its reconstruction accuracy in at least one of the clusters $C_{k}$ is at lease $\beta$ times $f(P, C)$ which is the expected accuracy obtained when using all of $P_{j}$ 's visible images $V_{j}$, i.e. is covered if:

$$
\max f\left(P_{j}, C_{k} \cap V_{j}\right) \geq \beta\left(P_{j}, V_{j}\right)
$$

Where $\lambda=0.7$ in the experiments. And $V_{j}$ is the visibility of the point $P_{j}$.

| Iteration | Intrinsic parameters to be estimated | Assumptions | Output Calibration Matrix |
| :---: | :---: | :---: | :---: |
| 1 | The focal length | $\begin{gathered} s=0 \\ \left(u_{0}, v_{0}\right)=\left(u_{c}, v_{c}\right) \\ r=\frac{f_{x}}{f_{y}}=1 \end{gathered}$ | $K_{1}=\left(\begin{array}{ccc}\tilde{f}_{x}^{(0)} & 0 & u_{c} \\ 0 & \tilde{f}_{x}^{(0)} & v_{c} \\ 0 & 0 & 1\end{array}\right)$ |
| 2 | The aspect ratio | $\begin{gathered} s=0 \\ \left(u_{0}, v_{0}\right)=\left(u_{c}, v_{c}\right) \end{gathered}$ <br> Use $K_{1}$ as initial guess | $K_{2}=\left(\begin{array}{ccc}\tilde{f}_{x}^{(1)} & 0 & u_{c} \\ 0 & \tilde{f}_{y}^{(1)} & v_{c} \\ 0 & 0 & 1\end{array}\right)$ |
| 3 | The principal point | $s=0$ <br> Use $K_{2}$ as initial guess | $K_{3}=\left(\begin{array}{ccc}\tilde{f}_{x}^{(2)} & 0 & \tilde{u}_{0}^{(0)} \\ 0 & \tilde{f}_{y}^{(2)} & \tilde{v}_{0}^{(0)} \\ 0 & 0 & 1\end{array}\right)$ |
| 4 | The focal lengths $f_{x}$ and $f_{y}$ | $s=0$ <br> Use $K_{3}$ as initial guess | $K_{4}=\left(\begin{array}{ccc}\tilde{f}_{x}^{(3)} & 0 & \tilde{u}_{0}^{(1)} \\ 0 & \tilde{f}_{y}^{(3)} & \tilde{v}_{0}^{(1)} \\ 0 & 0 & 1\end{array}\right)$ |
| 5 | Refinement of all parameters | $s=0$ <br> Use $K_{4}$ as initial guess | $K_{5}=\left(\begin{array}{ccc}\tilde{f}_{x}^{(4)} & 0 & \tilde{u}_{0}^{(2)} \\ 0 & \tilde{f}_{y}^{(4)} & \tilde{v}_{0}^{(2)} \\ 0 & 0 & 1\end{array}\right)$ |

Table 5. 1.Algorithm outline

## 5. Experiments

We have tested the proposed method on different datasets which were obtained from different databases on the web; we present here the results obtained from 2 images sequences.


Figure 5.1. The MVS clustering algorithm takes images, camera intrinsic parameters and poses estimation as input to generate the 3D model of the scene

Firstly we used a set of images, with low resolution, of the Old Town Hall [24] (figure 6.2). Secondly, a set of images with high resolution of the Topoi Lion scene [22] (figure 6.3) was used to confirm the good results obtained with the first set.

As input for the projective calibration phase; features had been detected and matched across all the input images. Based on the computed matches, the projective reconstruction was retrieved using the algorithm described in section 4 . Next, we used the computed projection matrices and projective 3D points as input guess for the non-linear optimization procedure described in section 5 to get the camera intrinsic parameters. Finally, using the computed intrinsic parameters, the dense 3D point cloud model of the scene is extracted using the Multi-View System described in the previous section.

The presented results in table 6.1 are the final results obtained after the fifth iteration of the multi-stage algorithm presented in table 5.1.


Figure 6.2. Some of the Old Town Hall images used for the test of the algorithm

|  |  | $\mathrm{f}_{\mathrm{x}}$ | $\mathrm{f}_{\mathrm{y}}$ | $\mathrm{u}_{0}$ | $\mathrm{v}_{0}$ |
| :--- | :--- | :--- | :--- | :--- | :--- |
| Frame 1 | Town | 2404.5 | 1121.6 | -2114.3 | -303.5 |
|  | Lion | 6.6610 | 6.4993 | -0.3773 | -2.285 |
| Frame2 | Town 1 | 2566.1 | 909.1 | -1158.6 | 1917 |
|  | Topoi | 5587 | 5.2419 | 0.0017 | -2.356 |
| Frame 3 | Town | 2567.1 | 1308.9 | -2718.8 | -381.8 |
|  | Topoi | 5.3842 | 4.6061 | -0.8287 | -3.046 |
| Frame 4 | Town | 2420.5 | 1123.2 | -2118.2 | -314.2 |
|  | Topoi | 5.9791 | 5.4461 | -0.4134 | -2.963 |

Table 6.1. The recovered focal lengths and principal point coordinates for the first four images of the sequence

The 3D points clouds of both scenes are presented in figures 6.4 and 6.5 respectively.
Software in [24] uses the computed camera parameters to generate the equivalent set oriented points, do the bundle adjustment of the whole data and finally output a polygon file containing the 3D structure of the scene. We used Meshlab to display the polygon file.

For the Topoi Lion scene, 122535 vertices had been obtained and 32286 vertices for the Old House Town scene.


Figure 6.3. Some of Topoi Lion images used for the test of the algorithm
148 Journal of Information \& Systems Management Volume 2 Number 3 September 2012

## 7. Conclusion and Perspectives

In this paper, we have presented a complete solution for the projective and metric upgrade of camera auto-calibration problem. Our algorithm deals with the case of varying intrinsic parameters and is consequently convenient for usage with the zooming/ focusing cameras.

For the success of the auto-calibration process and to ensure convergence to a global minimum at the end of the minimization process of equation (20), it's of utmost important to have an accurate initial guess at hand during the initialization stage of the optimization process of equation (20). To this end, a robust projective calibration algorithm was introduced that accurately estimate the necessary entities for the initialization stage.

In the experiments, we observed that the extraction of the 3D structure of the scene takes a long time and uses a lot of memory, mainly when using high resolution images. One may think in near future to deal with those problems by exploiting nowadays capacities of processors and memories. For this reason, we plan to implement the algorithm using the parallel programming concept. This will clearly reduce the execution time of the algorithm, and allow the use of high resolution images to get a denser model.


Figure 6.4. The recovered 3D dense structure for the Topoi Lion scene


Figure 6.5. The recovered 3D dense structure of the Old Town Hall scene

## References

[1] Triggs, B. (1996). Factorization Methods for Projective Structure and Motion, In: Proc. Int'l Conf. Computer Vision and Pattern Recognition, p. 845-851.
[2] Oliensis, J., Hartley, R. (2007). Iterative extensions of the Sturm/Triggs algorithm: Convergence and nonconvergence, IEEE Trans. Pattern Anal. Mach. Intell. 29 (12) 2217-2233.
[3] Oliensis, J., Genc, Y. (2001). Fast and Accurate Algorithms for Projective Multi-Image Structure from Motion, IEEE Trans. Pattern Analysis and Machine Intelligence, 23 (6) 546-559, June.
[4] Berthilsson, R., Heyden, A., Sparr, G. (1997). Recursive Structure and Motion from Image Sequences Using Shape and Depth Spaces, In: Proc. Int'l Conf. Computer Vision and Pattern Recognition, p. 444-449.
[5] Heyden, A., Berthilsson, R., Sparr, G. (1999). An Iterative Factorization Method for Projective Structure and Motion from Image Sequences, Image and Vision Computing, 17, 981-991.
[6] Mahamud, S., Hebert, M. (2000). Iterative Projective Reconstruction from Multiple Views, In: Proc. Int'l Conf. Computer Vision and Pattern Recognition, 2, 430-437.
[7] Hartley, R., Zisserman, A. (2004). Multiple View Geometry in Computer Vision, Cambridge University Press.
[8] Martin, A. (1981). Fischler and Robert C. Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography, Comm. of the ACM, 24 (6) 381-395, June.
[9] Wang, G., Wu, J. (2009). Perspective 3D Euclidean reconstruction with varying camera parameters, IEEE Trans. Circuits Syst. Video Technol. 19 (12) 1793-1803.
[10] Pollefeys, M., Van Gool, L. (1999). Stratified self-calibration with the modulus constraint, IEEE Trans. PAMI, 21(8), August, p. 707-724.
[11] Szlisky, R. (2011). Computer Vision Algorithms and Applications, Springer.
[12] Ze-tao, J., Shan-chao L. (2011). The Self-calibration of Varying Internal Camera Parameters Based on Image of Dual Absolute Quadric Transformation, INFORMATION AND AUTOMATION, Communications in Computer and Information Science, Springer, 86, 452-461.
[13] Triggs, B. (1997). Autocalibration and the absolute quadric, In: Proc. Conf. CVPR, Puerto Rico, p. 609-614, June.
[14] Lourakis, M. A., Deriche, R. (2000). Camera self-calibration using the Kruppa equations and the SVD of the fundamental matrix: The case of varying intrinsic parameters, Technical Report RR-3911, INRIA, France, March.
[15] Saaidi, A., Halli, A., Tairi, H., Satori, K. (2008). Self Calibration Using A Particular Motion Of Camera, WSEAS Transaction on Computer Research. 3 (5) May.
[16] Chandraker, M., Agarwal, S., Kahl, F., Nistér, D., Kriegman, D. (2007). Autocalibration via rank-constrained estimation of the absolute quadric. In: IEEE conference on computer vision and pattern recognition.
[17] Pollefeys, M., Khoch, R., Van Gool, L. (1998). Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters, In: Proc. 6th Int. Conf. on Computer Vision, Bombay, India, January, p. 90-95.
[18] E L-Attar, A., Tairi, H., Karim, M. (2011). Reconstruction 3D: Un Algorithme Hybride pour l'Amélioration des Performance de la Factorisation Projective, International Workshop on Information Technologies and Communication, WOTIC'11 - Oral. Casablanca. Morocco. September.
[19] E L-Attar, A., Karim, M., Tairi, H., Ionita, S. (2011). A Robust Multistage Algorithm for Camera Self-Calibration Dealing with Varying Parameters, Journal of Theoretical and Applied Information Technology. 32 (1) 46-54.
[20] Noah Snavely, Steven, M., Seitz, Richard Szeliski. (2006). Photo Tourism: Exploring image collections in 3D. ACM Transactions on Graphics (Proceedings of SIGGRAPH 2006).
[21] Furukawa, Y., Ponce, J. (2010). Accurate, Dense, and Robust Multi-View Stereopsis, IEEE Transactions on Pattern Analysis and Machine Intelligence, 32 (8).
[22] https://server.topoi.hu-berlin.de/groups/bundlertools/wiki/fd0c9/attachments/00511/TopoiLion.zip
[23] Banerjee, A., Krumpelman, C., Ghosh, J., Basu, S., Mooney, R. J. (2005). Model-based overlapping clustering. In: SIGKDD, p. 532-537. ACM.
[24] OSP, (2010), http://opensourcephotogrammetry.blogspot.com/

