# FLEXIBLE 3D RECONSTRUCTION SOFTWARE TOOL

**Tomáš Babinec**
Doctoral Degree Programme (1), FEEC BUT
E-mail: xbabin01@stud.feec.vutbr.cz

Supervised by: Ilona Kalová
E-mail: kalova@feec.vutbr.cz

## ABSTRACT

Purpose of the new software environment presented in this paper is to enable flexible solution of various 3D reconstruction tasks from image data by a single computing tool. This approach combines computational power of several independent programs which are based on slightly different algorithms but solve related sets of equations.

## 1. INTRODUCTION

Image based 3D shape reconstruction is a discipline which falls in the field of Computer vision. Its task is to regain the depth (shape) related information about measured objects in the scene via processing sets of obtained 2D images. The techniques of 3D reconstruction provide in general exceptionally beneficial ways for evaluation objects' and scenes' three-dimensional properties. Their power, in contrast to the traditional means of shape measurements, lies in the ability to almost instantly record the whole area of interest with no need for physical contact with the scanned objects. This enables the possibility to collect consistent data under fast changing conditions. In addition the obtained images can be easily stored for further analysis in the future.

Due to rapid development of microelectronics and increase of computational power, it is now feasible to design and implement vision systems capable of fast 3D measurements, which provide outstanding accuracy. These systems can be very compact with low demands on operating staff.

There is a wide range of different approaches to image based shape evaluation (see [1]). The software tool described in this paper is oriented to active and passive triangulation, which are nowadays the most commonly used methods. In general several different but closely related photogrammetric tasks have to be solved in order to achieve successful 3D reconstruction via triangulation techniques. Examples of such tasks can be camera calibration, optical distortion identification, 3D to 2D transformation and 3D point coordinates reconstruction.

Because the above mentioned photogrammetric problems are described by different equation sets, usually specialized programs are used for solving them. This approach however prevents the evaluation of possible relations between the separate tasks. As a result new application concept is being presented in this paper. It was designed with the attention to flexible definition of the photogrammetric tasks and the ability to simultaneously solve sets of different types of equations. This approach follows the work which was originally introduced in [2].

## 2. APPLICATION DESIGN

The created software environment consists of 3 main modules (see Figure 1). These are MANAGER, CALCULATER and I/O unit. The most important module was entitled MANAGER. It defines a basic layer for data processing. It is responsible for photogrammetric data storage and administration, compiles feasible equations according to given data dependences and initiates solution of unknown parameters in the CALCULATOR module.

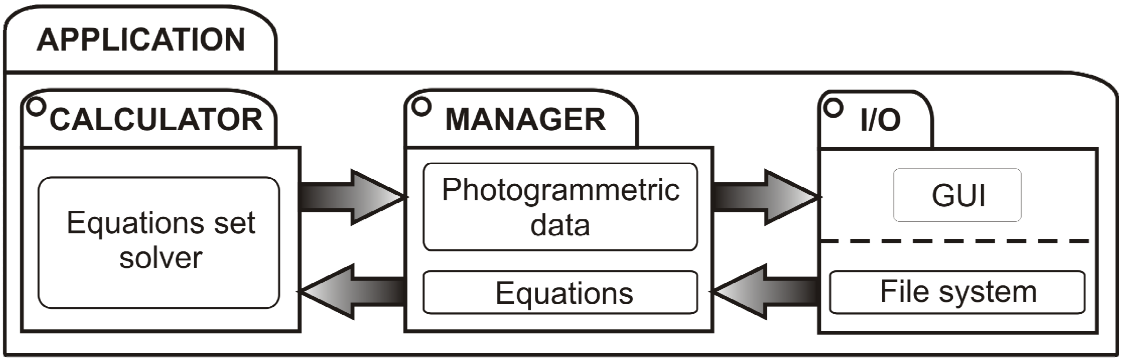


**Figure 1:** Application scheme.

Because the generated equations are usually highly nonlinear, the CALCULATOR has to implement some optimization algorithm which is capable of solving nonlinear problems. Data entry and results interpretation is done through I/O unit. It utilizes GUI (Graphical User Interface) to communicate directly with the user and enables current data to be stored as a text file.

### 2.1. DATA TYPES

For effective data processing and administration in *manager* a special data hierarchy had to be developed and implemented. The basic structural element is called *cell*. It describes one parameter of one scene object. Such a parameter is for example x-coordinate of a 3D point in the scene. The *cell* contains information about the value of the parameter, a measure of its uncertainty, limits of possible values, a step used in iterative computation, and a flag denoting the state of the parameter. The possible states of the flag are unknown, measured and constant.

Higher structures called *cluster* are composed from the *cells*. Every *cluster* type represents one type of a scene object. As it is shown in the Figure 2, there are now 8 different types of *clusters* possible to be defined.

- Distortion has 5 *cell*s for values of polynomial coefficients $a_1$, $a_2$, $a_4$, $a_6$, $a_8$ describing geometrical distortion of the camera. This way it is possible to express barrel and pincushion distortion together with the distortion caused by image sensor shift and skew.
- Camera has 4 *cell*s with interior parameters of the camera. These are the 2D projection center coordinates ($u_0$, $v_0$), camera constant $c$ and pixel width/height ratio $\frac{w}{h}$.

- Picture has 0 *cells*. Its main purpose is to express the connection between camera and appropriate view, which were employed when acquiring the image data.
- View has 6 *cells* with values of external camera parameters defining the camera position $(x, y, z)$ and orientation $\omega, \varphi, \kappa$ in 3D space.
- 2D, 3D point have 2/3 *cells* with appropriate point coordinates.
- Plane has 3 *cells* defining its position in 3D space.
- Line has 4 *cells* defining its position in 3D space.

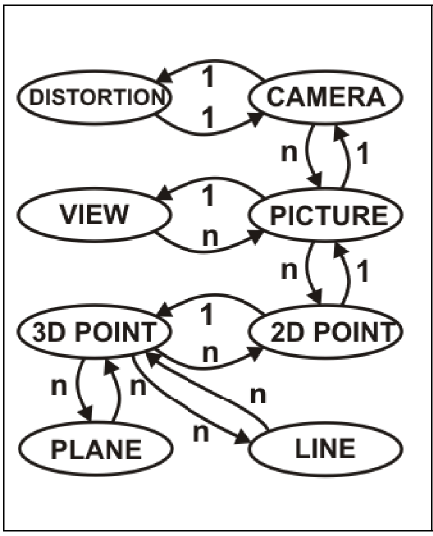


**Figure 2:** CLUSTERs.

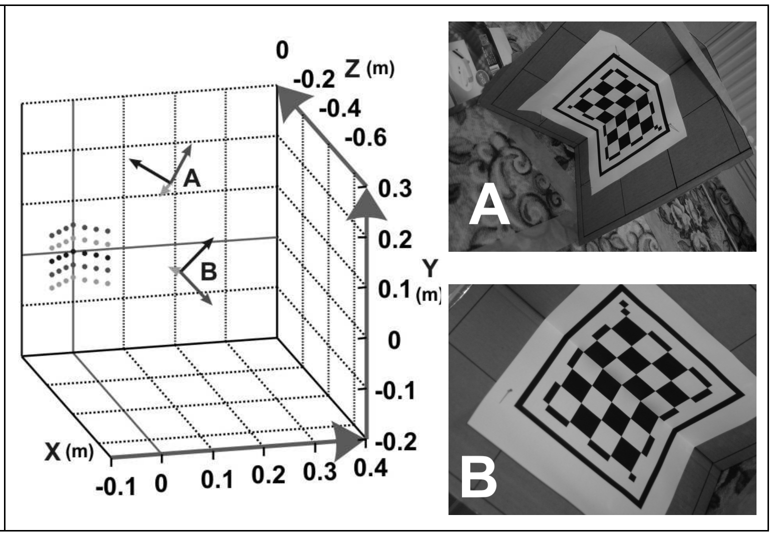


**Figure 3:** Real calibration.

Besides the values of scene objects' parameters the applications also requires an input of existing relationships and dependencies among scene objects. In the software environment it is represented by connections between appropriate *clusters*. Possible connections are depicted by arrows in Figure 2. Every arrow symbolizes a feasible link. The adjacent symbol indicates the number of possible connections.

The scheme for example expresses the fact, that the particular picture could have been captured by only one camera, which was positioned in one particular view at the scene. However one camera could have captured several different pictures, each from a different view.

### 2.2. EQUATIONS SET SOLVER

As was motioned before the generated equations are generally nonlinear. Algorithms suitable for their solution (e.g. gradient based methods, genetic algorithms, differential evolution or modifications of least squares method) have an iterative character.

At present iterative adjustment of the parameters by the least squares method [3] is used. Utilization of this algorithm has already been verified in the original work [2]. It operates with $N$ functions $F(\mathbf{x}, \mathbf{z})$, which describe photogrammetric relations in a form of $F_i(\mathbf{x}, \mathbf{z}) = 0$, where $i = 1, 2, \ldots, N$, $\mathbf{z} = [z_1, z_2, \ldots, z_P]^T$ is a vector of $P$ measured values and $\mathbf{x} = [x_1, x_2, \ldots, x_M]^T$ denotes vector of $M$ unknown parameters. The goal of this method is to iteratively optimize unknown parameters in vector $\tilde{\mathbf{x}}$ and measured values in $\tilde{\mathbf{z}}$ to satisfy equations (1) under the condition (2).

$$\mathbf{F}(\tilde{\mathbf{x}}, \tilde{\mathbf{z}}) = 0 \tag{1}$$

$$\left|(\tilde{\mathbf{z}} - \mathbf{z})^T \mathbf{K}_0^{-1} (\tilde{\mathbf{z}} - \mathbf{z})\right| = \min \tag{2}$$

$\mathbf{K}_0$ is a covariance matrix for the vector of measured parameters $\mathbf{z}$.

Application of the Lagrange multipliers method [3] (minimizing the auxiliary function) on the equations (1) and (2) leads to the computational expressions for the correction vectors $\mathbf{\Delta x}$ and $\mathbf{\Delta z}$. In each step of the method values of the unknown parameters are updated using the corrections $\mathbf{\Delta x}$ according to $\tilde{\mathbf{x}}_{k+1} = \tilde{\mathbf{x}}_k + \Delta\mathbf{x}$. Iiteration should continue until the values in correction vector $\mathbf{\Delta x}$ are sufficiently low .Vector $\mathbf{\Delta z}$ then contains corrections to the measured values, which will lead to further minimization of the total error.

| | Initial conditions | Correct values | 1 | 2 | 3 | Without distortion | Absolute Error Δ | Relative error δ (%) |
|---|---|---|---|---|---|---|---|---|
| $u_0$ (p) | 400,00 | 400,00 | 400,00 | 400,00 | 400,00 | 346,29 | 53,71 | 13,4 |
| $v_0$ (p) | 300,00 | 300,00 | 300,00 | 300,00 | 300,00 | -118,31 | 418,31 | 139,4 |
| c (p) | 200,00 | 1000,00 | 323,85 | 468,01 | 1000,00 | 509,83 | 490,17 | 49,0 |
| w/h | 0,80 | 1,00 | 1,01 | 1.00598 | 1,00 | 0,68 | 0,32 | 31,6 |
| x (J) | -2,00 | 0,00 | 0,16 | 0,16 | 0,00 | 0,02 | -0,02 | - |
| y (J) | 5,00 | 1,00 | 0,77 | 0,77 | 1,00 | 1,06 | -0,06 | -6,0 |
| z (J) | -6,00 | -4,00 | -2,05 | -2,05 | -4,00 | -4,04 | 0,04 | -0,9 |
| ω (°) | 40,00 | 10,00 | 17,20 | 17,20 | 10,00 | 60,29 | -50,29 | -502,9 |
| φ (°) | 20,00 | 0,00 | -8,04 | -8,04 | 0,00 | -6,03 | 6,03 | - |
| κ (°) | -20,00 | 0,00 | 1,60 | 1,60 | 0,00 | 2,07 | -2,07 | - |
| a1 (-) | 6,00 | 1,00 | 6,00 | 0,22 | 1,00 | 0,00 | 0,00 | - |
| a2 (-) | -5,00 | -0,003 | -5,00 | 0,00 | 0,00 | 0,00 | 0,00 | - |

**Table 1:** Camera calibration combined with geometrical distortion identification.

## 3. RESULTS/FUNCTION VERIFICATION

Several photogrammetric tasks have been simulated using the MatLab environment to verify the functionality of created software tool. These task included camera calibration and 3D point coordinates reconstruction. Table 1 contains results obtained via camera calibration combined with geometrical distortion identification, which well illustrates the advantages of the application's flexibility.

The most right third of the table shows the outcome of camera calibration based on a distorted image if the camera distortion is neglected. In such a case the parameters errors are rather significant. On the other hand, the middle third of the table 1 corresponds to calibration combined with camera distortion. Since the external and internal camera parameters can counteract their effect with distortion model, the best way to find the correct values of the unknown parameters is to divide the calibration process into three steps.

First a rough calibration neglecting distortion is done. This serves to estimate external camera parameters. In the second step the exterior parameters are frozen, and an estimate of camera distortion is calculated. In the last step all parameters are marked as unknown to be optimized. This strategy leads to expected correct results.

Real data camera calibration test (see Figure 3) had been conducted and the results were verified using a reconstruction environment described in [4].

## 4. CONCLUSION AND FUTURE WORK

The described software tool for 3D reconstruction was implemented in C/C++ with emphasis to possible future augmentation. It enables simultaneous solution of versatile photogrammetric tasks. Behavior of the application was tested on synthetic and real data with satisfactory results. Next step in the application development will be an expansion of computational abilities and automation of the data processing.


## ACKNOWLEDGEMENT

This work has been supported in part by Ministry of Education, Youth and Sports of the Czech Republic (Research Intent MSM0021630529 Intelligent systems in automation (VZ UAMT), Grant Agency of the Czech Republic (102/09/H081 SYNERGY - Mobile Sensoric Systems and Network) and by Brno University of Technology. Without kind support of the above-mentioned agencies and institutions the presented research and development would not be possible.


## REFERENCES


[1] Jähne B., Haußecker H., Geisler P.: *Handbook of Computer Vision and Applications, Volume 1 – 3,* San Diego: Academic Press, 1999. ISBN 0-12-379770-5

[2] Popov P., Richter M.: *Exterior orientation of digital images using CCD cameras in close-range photogrammetry*. Výzkumná zpráva pro GACR 101/93/2435 grantu „Advanced schemes of robot control“, Brno, 1995

[3] Тюфлин Ю. С. *Способы стереофотограмметрической обработки снимков, полученных с подвижного базиса*. Москва: Недра, 1971.

[4] Babinec, T. *Vyhodnocení tvaru objektů.* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2007. 66 s. Vedoucí bakalářské práce Ing. Miloslav Richter, Ph.D.